Word Confidence Measures for Machine Translation

Von der Fakultät für Mathematik, Informatik
und Naturwissenschaften der
Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades einer
Doktorin der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Mathematikerin

Nicola Ueffing

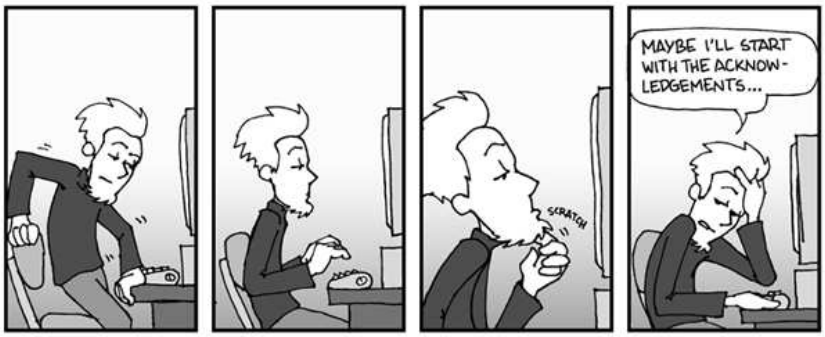
aus Aachen

Berichter: Universitätsprofessor Dr.-Ing. Hermann Ney
Professor Dr. Enrique Vidal

Tag der mündlichen Prüfung: 15. März 2006

To Markus and my family

Acknowledgments



"Piled Higher and Deeper" by Jorge Cham (www.phdcomics.com)

First, I would like to express my gratitude to my advisor Professor Dr.-Ing. Hermann Ney, head of the Lehrstuhl für Informatik VI at the RWTH Aachen University. This thesis would not have been possible without his advice and the very interesting work environment in his group. I am grateful that he offered me the possibility to participate in the CLSP summer workshop 2003.

I would also like to thank Professor Dr. Enrique Vidal, from the Departamento de Sistemas Informáticos Y Computación at the Universidad Politecnica de Valencia, for agreeing to review this thesis and for for his interest in this work.

Special thanks go to Thomas Schoenemann and Gregor Leusch for their valuable work on HMM based confidence estimation and on automatic evaluation for MT. All the other people at the Lehrstuhl für Informatik VI also deserve my gratitude for many fruitful discussions, helpful feedback, and for the very good working atmosphere. They also provided very good "non-working atmosphere" in the coffee breaks, the sports events, and the other recreational activities. Especially the "Geigeltruppe" and the "cool guys" provided lots of fun. I want to thank all those who helped me when writing this thesis by proofreading it, introducing me to nicer ways of writing formulae, and making me think about the ligatures in "Ueffing".

I would like to say "thanks a bunch" to all the people who made the CLSP summer workshop on Confidence Estimation for Statistical Machine Translation possible: Fred Jelinek and his team at CLSP for organizing the workshop. George Foster and Simona Gandrabur for being inspiring, engaged, conscientious, and funny team leaders and for keeping in touch after the workshop. John Blatz, Erin Fitzgerald, Cyril Goutte, Alex Kulesza, and Alberto Sanchis (in alphabetical order) for being a wonderful team which worked hard and had a great time together.

Großer Dank gilt meiner Familie für ihre Liebe und Unterstützung. Elsbeth, Klaus, Petra, Jörg, Karin und Josef waren immer für mich da, haben mir Geborgenheit vermittelt und waren immer für einen guten Ratschlag und Spaß zu haben.

Besonders möchte ich mich bei Markus Beermann bedanken für seine Liebe und Aufmunterung, sowie seine nahezu unerschütterliche westfälische Gelassenheit. Er hat mich motiviert und unterstützt, aber mir auch stets geholfen, auf dem Boden zu bleiben und das Leben außerhalb der Arbeit nicht zu vergessen und es zu genießen.

This thesis is based on work carried out during my time as a research scientist at the Department for Computer Science at the RWTH Aachen University, Germany. The work was partially funded by European Union under the RTD project TransType2 (IST–2001–32091), the integrated project TC-STAR – Technology and Corpora for Speech to Speech Translation (IST-2002-FP6-506738), and by the Deutsche Forschungsgemeinschaft (DFG) under the project "Statistical Methods for Written Language Translation" (Ne572/5).

Abstract

Due to continuous research which led to improved concepts and algorithms, the quality of automatically generated translation has significantly improved in recent years. However, the performance of machine translation systems is still not perfect. For human users dealing with these systems, it is desirable to obtain a reliable indication of possible errors in the system output. The same holds for applications based on machine translation technologies. They could explore the knowledge about possible mistakes. Interactive machine translation systems, for example, can modify or even discard predicted translations which are identified as possible errors. The aim of this work is to provide knowledge about when a translation generated by the system is incorrect by calculating measures of confidence for each word in this translation. This topic has hardly been investigated in machine translation before. Different ways of determining confidence measures are proposed and experimentally evaluated in this thesis. The basic concept behind all these approaches are word posterior probabilities. The goal is to set up a sound theoretical framework for the calculation and evaluation of confidence measures in machine translation.

The main problem which has to be solved for the computation of word posterior probabilities is to define the underlying concept. There exists no intuitive definition of this concept. Possible approaches include the word posterior probability of a word based on its position in the sentence and the occurrence in any position. Several solutions to this problem are presented in this thesis. Furthermore, different approaches to the calculation of word posterior probabilities are introduced and compared. They can be divided into two categories: system-based methods which explore knowledge provided by the translation system that has generated the translations, and direct methods which are independent of the translation system. The system-based techniques make use of system output, such as word graphs or N-best lists. The word posterior probability is determined by summing the probabilities of all sentences which contain the target word. The direct confidence measures developed here take other knowledge sources, such as word or phrase lexica, into account. They can be applied to the output of non-statistical machine translation systems as well.

The word posterior probabilities introduced in this thesis can directly be applied as confidence measures as follows: For a given translation generated by a machine translation system, the posterior probabilities of all words are determined and compared to a threshold. All words whose posterior probability is above this threshold are tagged as correct and all others are tagged as incorrect. To evaluate the proposed confidence measures, the information on which words are correct is needed. In machine translation, it is not intuitively clear how to determine the correctness of single words. As a solution to this problem, several different ways of deriving word error measures from existing machine translation evaluation metrics are suggested and investigated. The relation between the word error measure and the word posterior probabilities is studied in detail. From the formulation of the posterior risk for different error measures, a theoretical foundation of the word posterior probabilities is derived.

The different confidence measures proposed here explore information from various knowledge sources, such as sentence probabilities provided by the machine translation

system and statistical word and phrase lexica. To explore the knowledge from all these sources, a combination of several confidence measures is investigated.

The suggested methods are evaluated on different translation tasks and several language pairs. In order to assess the general discriminative power of the confidence measures, they are tested on output from four different machine translation systems. Three of those are state-of-the-art phrase-based systems, and the fourth is an established rule-based system. A significant improvement in terms of confidence error rate is achieved in all settings.

In this work, applications of confidence measures that improve translation quality of state-of-the-art systems are investigated. These include rescoring with confidence measures and their use in an interactive machine translation system. Rescoring with confidence measures is shown to improve translation quality. In the interactive machine translation environment, word confidence measures are successfully applied to select and discard possible translations based on their confidence. For the evaluation of the interactive translation experiments, an existing automatically determined metric is used. An extension to this metric is proposed to better model the gain achieved from using the system in a real-world application. The experiments show that the quality of the predicted translations is improved through the use of confidence measures.

Zusammenfassung

Infolge verbesserter Konzepte und Algorithmen hat sich die Qualität von automatisch erzeugten Übersetzungen in den letzten Jahren deutlich verbessert. Dennoch ist die Leistung solcher Übersetzungssysteme noch lange nicht perfekt. Für die Weiterverarbeitung von automatisch erzeugten Übersetzungen ist es wünschenswert zu wissen, wann das System fehlerhafte Übersetzungen ausgibt. Dies gilt sowohl für Übersetzungssysteme, die von Menschen genutzt werden, als auch in Anwendungen, in denen die automatische Übersetzung eine Vorstufe für weitere Sprachverarbeitung darstellt. Beispielsweise können interaktive Übersetzungssysteme Teile der Übersetzung modifizieren oder sogar ganz verwerfen, wenn diese mit hoher Wahrscheinlichkeit Fehler enthalten. Das Ziel dieser Arbeit ist es, das Wissen über mögliche Fehler bereitzustellen. Dazu werden Konfidenzmaße für die einzelnen Wörter in der Systemausgabe berechnet. Dieses Thema wurde für die automatische Übersetzung bisher kaum behandelt. Alle hier vorgestellten Verfahren basieren auf Wortposteriorwahrscheinlichkeiten. Das Ziel ist, einen theoretischen Rahmen für die Berechnung und die Bewertung von Konfidenzmaßen für die automatische Übersetzung zu definieren.

Das Hauptproblem ist die Definition des zugrundeliegenden Konzepts für die Wortposteriorwahrscheinlichkeiten. In der maschinellen Übersetzung existiert keine intuitive Definition eines solchen Konzepts. Mögliche Ansätze umfassen Wortposteriorwahrscheinlichkeiten basierend auf der Position des Wortes im Satz oder dem Auftreten in einer beliebigen Position. Verschiedene Lösungen dieses Problems werden diskutiert und experimentell bewertet. Des weiteren werden verschiedene Methoden zur Berechnung von Wortposteriorwahrscheinlichkeiten eingeführt und verglichen. Sie lassen sich in zwei verschiedene Kategorien einteilen: Systembasierte Ansätze, die vom Übersetzungssystem zur Verfügung gestellte Informationen ausnutzen, und direkte Modelle, die unabhängig von diesem System sind. Die systembasierten Techniken arbeiten auf der Basis von Systemausgaben wie Wortgraphen oder N-Best Listen. Die Wortposteriorwahrscheinlichkeiten werden durch Summierung der Satzwahrscheinlichkeiten all jener Sätze bestimmt, die das betrachtete Wort enthalten. Die hier entwickelten direkten Konfidenzmaße nutzen andere Wissensquellen wie z.B. statistische Wort- und Phrasenlexika. Diese Maße können auch auf Übersetzungen angewendet werden, die von nicht-statistischen Systemen erzeugt wurden.

Die in dieser Arbeit eingeführten Wortposteriorwahrscheinlichkeiten können direkt als Konfidenzmaße verwendet werden: Für jedes Wort in einer automatisch generierten Übersetzung wird die Wortposteriorwahrscheinlichkeit berechnet und mit einem Schwellwert verglichen. Alle Wörter, deren Posteriorwahrscheinlichkeit über diesem Schwellwert liegt, werden als korrekte Übersetzungen akzeptiert, und alle anderen werden verworfen. Um Konfidenzmaße in diesem Szenario zu bewerten, müssen die tatsächlichen Wortfehler bekannt sein. In der maschinellen Übersetzung ist jedoch nicht intuitiv klar, wie Fehler auf Wortebene bestimmt werden können. Deshalb werden in der vorliegenden Arbeit mehrere mögliche Lösungen dieses Problems vorgestellt und experimentell verglichen. Sie basieren auf verschiedenen etablierten Bewertungsmaßen für die automatische Übersetzung. Der Zusammenhang zwischen dem Wortfehlermaß und Wortposteriorwahrscheinlichkeiten wird detailliert untersucht. Zu diesem Zweck wird

das Bayessche Risiko für verschiedene Fehlermaße formuliert und damit eine theoretische Grundlage für die Wortposteriorwahrscheinlichkeiten geschaffen.

Die in dieser Arbeit vorgeschlagenen Konfidenzmaße nutzen Informationen aus unterschiedlichen Wissensquellen. Diese umfassen die vom Übersetzungssystem berechneten Satzwahrscheinlichkeiten und statistische Wort- und Phrasenlexika. Um die Informationen aus diesen verschiedenen Quellen zu verbinden, wird die Kombination von mehreren Konfidenzmaßen untersucht.

Die vorgestellten Methoden werden auf verschiedenen Übersetzungsaufgaben und mehreren unterschiedlichen Sprachenpaaren experimentell untersucht. Um ein möglichst umfassendes Bild von der Qualität der Konfidenzmaße zu erhalten, werden sie auf die Ausgabe von vier unterschiedlichen Übersetzungssystemen angewendet. Drei dieser Systeme sind aktuelle statistische Systeme, und das vierte ist ein etabliertes Regelbasiertes. Die Konfidenzfehlerrate kann in allen untersuchten Szenarien signifikant gesenkt werden.

In der vorliegenden Arbeit werden ferner Anwendungen von Konfidenzmaßen untersucht, die die Qualität maschineller Übersetzungen verbessern. Diese Methoden umfassen Rescoring mit Konfidenzmaßen sowie ihre Anwendung in interaktiven Übersetzungssystemen. Es zeigt sich, dass das Rescoring die Qualität der Übersetzungen verbessert. In der interaktiven Übersetzungsumgebung werden Konfidenzmaße erfolgreich zur Auswahl und zur Verwerfung von möglichen Übersetzungen eingesetzt. Zur Bewertung dieser Experimente wird ein etabliertes automatisches Maß verwendet und erweitert, so dass es besser den Nutzen eines interaktiven Systems in einer realen Anwendung modelliert. Die Experimente zeigen, dass die Qualität der vom System vorgeschlagenen Übersetzungen durch den Einsatz von Konfidenzmaßen verbessert wird.

Auszug[a]

Wegen der ununterbrochenen Forschung, die zu verbesserte Konzepte und Algorithmen führen, hat die Qualität der automatisch erzeugten Übersetzung erheblich in den letzten Jahren verbessert. Jedoch ist die Leistung der maschineller Übersetzungsysteme noch nicht vollkommen. Für die menschlichen Benutzer, die diese Systeme beschäftigen, ist es wünschenswert, eine zuverlässige Anzeige über mögliche Störungen in der Systemausgabe zu erhalten. Die gleichen Einflüsse für die Anwendungen basiert auf Übersetzungtechnologien. Sie konnten das Wissen über mögliche Fehler erforschen. Wechselwirkende maschinelle Übersetzungsysteme z.B. können vorausgesagte Übersetzungen ändern oder sogar wegwerfen, die als mögliche Störungen gekennzeichnet werden. Das Ziel dieser Arbeit ist, Wissen ungefähr zur Verfügung zu stellen, wenn eine Übersetzung, die durch das System erzeugt wird, falsch ist, indem sie Masse Vertrauen für jedes Wort in dieser Übersetzung errechnet. Dieses Thema ist kaum in der maschinellen Übersetzung vorher nachgeforscht worden. Unterschiedliche Weisen der Bestimmung von von Vertrauen Massen werden vorgeschlagen und ausgewertet experimentell in diese These. Das Grundmodell hinter allen diesen Annäherungen sind Worthinterteilwahrscheinlichkeiten. Das Ziel ist, einen stichhaltigen theoretischen Rahmen für die Berechnung und die Auswertung der Vertrauen Masse in der maschinellen Übersetzung aufzustellen.

Das Hauptproblem, das für die Berechnung der hinteren Wahrscheinlichkeiten des Wortes gelöst werden muss, soll das zugrundeliegende Konzept definieren. Besteht keine intuitive Definition dieses Konzeptes. Mögliche Annäherungen schließen die hintere Wahrscheinlichkeit des Wortes eines Wortes ein, das auf seiner Position im Satz und dem Auftreten in jeder möglicher Position basiert. Einige Lösungen zu diesem Problem werden in dieser These dargestellt. Ausserdem werden unterschiedliche Annäherungen an die Berechnung der hinteren Wahrscheinlichkeiten des Wortes eingeführt und verglichen. Sie können in zwei Kategorien geteilt werden: System-gegründete Methoden, die Wissen erforschen, stellten auf das Übersetzung System, das die Übersetzungen erzeugt hat und die direkten Arten zur Verfügung, die Unabhängiges des Übersetzung Systems sind. Die System-gegründeten Techniken gebrauchen Systemausgabe, wie Wortdiagramme oder N-beste Listen. Die hintere Wahrscheinlichkeit des Wortes wird festgestellt, indem man die Wahrscheinlichkeiten aller Sätze summiert, die das Zielwort enthalten. Die direkten Vertrauen Masse, die hier entwickelt werden, nehmen andere Wissen Quellen, wie Wort oder Phrase lexica, in Betracht. Sie können am Ausgang der non-statistical maschineller Übersetzungsysteme außerdem angewendet werden.

Die hinteren Wahrscheinlichkeiten des Wortes, die in dieser These eingeführt werden, können direkt angewendet werden, während Vertrauen misst, wie folgt: Für eine gegebene Übersetzung, die durch ein maschinelles Übersetzungsystem erzeugt wird, werden die hinteren Wahrscheinlichkeiten aller Wörter mit einer Schwelle festgestellt und verglichen. Alle Wörter deren hintere Wahrscheinlichkeit über dieser Schwelle sind etikettiertes so korrektes ist und alle andere werden wie falsch etikettiert. Die vorgeschlagenen Vertrauen Masse auszuwerten, über die die Informationen Wörter korrekt sind ist erforderlich. In der maschinellen Übersetzung ist es nicht intuitiv frei, wie man die Korrektheit der einzelnen

[a] Translated from English into German by the Systran version available at `http://babelfish.altavista.com/tr` in November 2005.

Wörter feststellt. Als Lösung zu diesem Problem, werden einige unterschiedliche Weisen des Ableitens von von Wortstörung Massen von vorhandenen Übersetzung-Auswertung Metriken vorgeschlagen und nachgeforscht. Die Relation zwischen dem Wortstörung Maß und den hinteren Wahrscheinlichkeiten des Wortes wird im Detail studiert. Von der Formulierung der Bayes Gefahr für unterschiedliche Störung Masse, wird eine theoretische Grundlage der hinteren Wahrscheinlichkeiten des Wortes abgeleitet.

Die unterschiedlichen Vertrauen Masse, die hier vorgeschlagen werden, erforschen Informationen von den verschiedenen Wissen Quellen, wie Satzwahrscheinlichkeiten, die von vom maschinellen Übersetzungsystem und statistischen Wort und Phrase dem lexica bereitgestellt werden. Um das Wissen von allen diesen Quellen zu erforschen, wird eine Kombination einiger Vertrauen Masse nachgeforscht.

Die vorgeschlagenen Methoden werden auf unterschiedliche Übersetzung Aufgaben und einige Sprachpaare ausgewertet. Um die allgemeine unterscheidende Energie der Vertrauen Masse festzusetzen, werden sie auf Ausgang von vier unterschiedlichen maschinellen Übersetzungsystemen geprüft. Drei von denen sind state-of-the-art Phrase-gegründete Systeme, und das Viertel ist ein hergestelltes rule-based System. Eine bedeutende Verbesserung in der Vertrauen Fehlerhäufigkeit wird in allen Einstellungen erzielt.

In dieser Arbeit werden Anwendungen der Vertrauen Masse, die Übersetzung Qualität der state-of-the-art Systeme verbessern, nachgeforscht. Diese schließen das Rescoring mit Vertrauen Massen und ihren Gebrauch in einem wechselwirkenden maschinellen Übersetzungsystem ein. Rescoring mit Vertrauen Massen wird gezeigt, um Übersetzung Qualität zu verbessern. Im wechselwirkenden Übersetzungklima werden Wortvertrauen Masse erfolgreich, um die möglichen Übersetzungen vorzuwählen und wegzuwerfen angewendet, die auf ihrem Vertrauen basieren. Für die Auswertung der wechselwirkenden Übersetzung Experimente, automatisch stellte bestehen metrisches wird verwendet fest. Eine Verlängerung zu diesem metrischen wird vorgeschlagen, um Modell zu verbessern, das der Gewinn vom Verwenden des Systems in einer realistischen Anwendung erzielte. Die Experimente zeigen, dass die Qualität der vorausgesagten Übersetzungen durch den Gebrauch von Vertrauen Massen verbessert wird.

Contents

List of Tables

List of Figures

1 Introduction

Over the last years, machine translation (MT) technology has received growing interest leading to improved algorithms and concepts. However, the translations generated by MT systems still contain errors. The goal of this thesis is to identify these errors automatically and distinguish the parts of the translation which are correct from those which are incorrect.

In many applications of MT, information about the correctness of the automatically generated translations is useful. This is true in scenarios where human users are involved, but also if other natural language processing components like information extraction process the automatic translations. Examples of different tasks where machine translation is applied are

- complete and accurate translation of texts, e.g. technical manuals or parliamentary debates and other documents in multilingual organizations,
- a rough translation of a foreign language text, like a web page or news, that gives an idea of its contents,
- translation aid systems for human translators.

In all these contexts, it is important to know when the system possibly made an error, and when one can be sure of obtaining a good translation. The aim of this work is to provide this knowledge by calculating measures of **confidence** for the generated translations. Since often a translation of a sentence as a whole is incorrect, but contains correct parts, the confidence measures are determined for each word in the translation rather than for the whole sentence. Different ways of determining confidence measures will be investigated in this thesis. The basic concept behind all these approaches are **word posterior probabilities** which give an estimate of the probability of correctness of a word. Additionally, applications of confidence measures that improve system performance, e.g. through rescoring or in an interactive translation environment, will be presented.

After briefly reviewing the basic concept of statistical machine translation, this chapter will present the related work in the area of confidence estimation for machine translation. It will give an overview of the state of the art in sentence-level as well as word-level confidence estimation and the applications investigated so far within the (statistical) machine translation community.

1.1 Statistical machine translation

The statistical approach to machine translation has received growing interest over the last years since its introduction by the IBM research group in the early nineties. In various comparative evaluations, it has been proven competitive or superior to other "traditional" approaches. The translation quality achieved in restricted domains is relatively high. Examples include the domains of appointment scheduling, which was the scope of the project Verbmobil [Wahlster 00], or tourism which is used in the IWSLT evaluations [Akiba et al. 04a]. In recent years, more challenging tasks have been tackled in SMT research. The TC-STAR project [tcs 05], for example, deals with speech translation of the plenary sessions of the European Parliament. The domain and the vocabulary of these speeches are open. The automatic translation has to cope with effects of spontaneous speech, misrecognitions from the speech recognition engine, and unknown words or constructs which are due to the large domain.

The goal of machine translation is the automatic translation of a source language string $f_1^J = f_1 \dots f_j \dots f_J$ of words f_j into a target language string $e_1^I = e_1 \dots e_i \dots e_I$. In statistical machine translation (SMT), the translation is modeled as a decision process: Given a source string f_1^J, the target string e_1^I with maximal posterior probability is determined:

$$\begin{aligned} \hat{e}_1^{\hat{I}} &= \underset{I,e_1^I}{\operatorname{argmax}} \left\{ Pr(e_1^I \,|\, f_1^J) \right\} \\ &= \underset{I,e_1^I}{\operatorname{argmax}} \left\{ \frac{Pr(f_1^J \,|\, e_1^I) \cdot Pr(e_1^I)}{Pr(f_1^J)} \right\} \\ &= \underset{I,e_1^I}{\operatorname{argmax}} \left\{ Pr(f_1^J \,|\, e_1^I) \cdot Pr(e_1^I) \right\} \end{aligned} \tag{1.1}$$

Through this decomposition of the posterior probability $Pr(e_1^I \,|\, f_1^J)$, two knowledge sources are obtained: the translation model $Pr(f_1^J \,|\, e_1^I)$ and the language model $Pr(e_1^I)$. Both of them can be modeled independently of each other. The translation model is responsible of linking the source string f_1^J and the target string e_1^I, i.e. of capturing the semantics of the sentence. The target language model $Pr(e_1^I)$ assigns probabilities to target word sequences. It models the well-formedness or the syntax in the target language. The probability of the source sentence, $Pr(f_1^J)$, is usually omitted in the maximization because it does not affect the choice of the target word sequence. Nevertheless, it will be shown later that this probability is important for the methods suggested in this thesis. The overall architecture of the statistical translation approach is depicted in figure 1.1.

The correspondence between the words in the source and the target string is described by alignments which can be viewed as mappings $a : j \rightarrow a_j \in \{0, \dots, I\}$ assigning a target position a_j to each source position j [Brown et al. 93]. An artificial target position zero is introduced for mapping source words that do not have any equivalence in the target string. The alignment is introduced into the model as hidden variable:

$$Pr(f_1^J \,|\, e_1^I) = \sum_{a_1^J} Pr(f_1^J, a_1^J \,|\, e_1^I) \; . \tag{1.2}$$

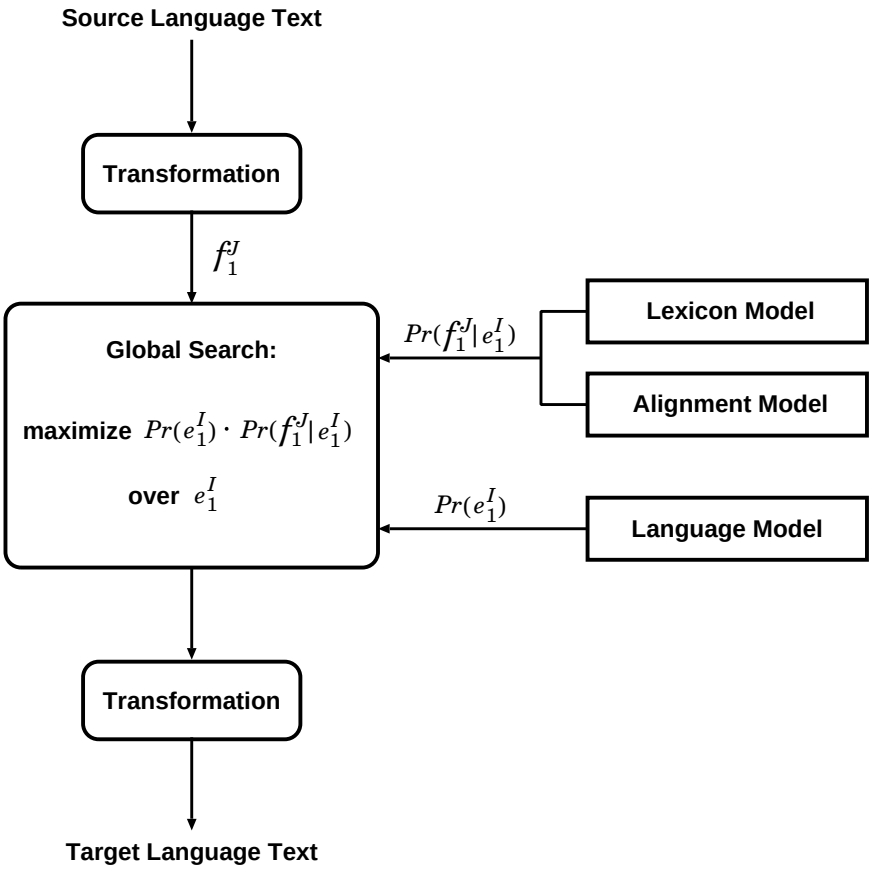


Figure 1.1: Architecture of the statistical approach to machine translation.

In state-of-the-art SMT systems as described below, these dependencies are often modeled implicitly through bilingual phrases.

The search (denoted by the argmax operation in equation 1.1) explores the space of all possible target language strings e_1^I and all possible alignments a_1^J between the source and the target language string to find the one with maximum probability.

Many current SMT systems pursue an alternative approach of modeling the translation process: the posterior probability is directly modeled as a log-linear combination of feature functions, $h_m(e_1^I, f_1^J)$, $m = 1, \ldots, M$, based on the maximum entropy framework [Och and Ney 04]. These functions assign a score to the sentence pair (e_1^I, f_1^J). The search criterion then results to

$$\begin{aligned} \hat{e_1^I} &= \underset{I, e_1^I}{\operatorname{argmax}} Pr(e_1^I \mid f_1^J) \\ &= \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m \cdot h_m(e_1^I, f_1^J) \right\} , \end{aligned}$$

where λ_m, $m = 1, \ldots, M$ are the interpolation weights. This allows for the integration of many different sub-models. Their weights can be directly optimized w.r.t. some MT evaluation criterion [Och 03]. The model given in equation 1.1 can be interpreted as a special case of this.

Most of the state-of-the-art SMT systems are based on bilingual phrases [Koehn et al. 03, Kumar and Byrne 03, Bertoldi et al. 04, Och and Ney 04, Sumita et al. 04, Vogel et al. 04]. Note that these phrases are sequences of words in the two languages and not necessarily phrases in the linguistic sense. Two phrase-based translation approaches will be described in more detail in section 3.3.3.1 and appendix C.1.

Another approach which has become popular in recent years is to incorporate syntactic knowledge into statistical MT systems [Yamada and Knight 02, Lin 04, Ding and Palmer 05]. These approaches parse the sentence(s) in one or both of the involved languages. The translation operations are then defined on parts of the parse tree(s). [Chiang 05] constructs hierarchical transducers for translation. The model is a syntax-free grammar which is learnt from a bilingual corpus without any syntactic information. It consists of phrases which can contain sub-phrases, so that a hierarchical structure is induced.

The third main approach which is currently investigated in SMT is the modeling of the translation process as a finite state transducer [Alshawi 96, Vidal 97, Bangalore and Riccardi 00, Casacuberta et al. 01, Kanthak and Ney 04, Mariño et al. 05]. This approach solves the translation problem by estimating a language model on a bilanguage defined over source and target language. The translation transducer is basically an acceptor for this bilanguage. More details on the RWTH system based on finite state transducers will be given in appendix C.2.

1.2 State of the art and related work in confidence estimation

1.2.1 Confidence estimation for machine translation

Confidence measures have been extensively studied for speech recognition [Weintraub et al. 97, Wessel et al. 01], but are not well known in other areas. Only recently, researchers have started to investigate confidence measures for machine translation [Blatz et al. 03, Gandrabur and Foster 03, Blatz et al. 04, Quirk 04, Sanchis 04]. This section will give an overview of confidence estimation for machine translation on the *word level* as well as the *sentence level* and discuss its applications.

The first work which studied confidence estimation for statistical machine translation was [Gandrabur and Foster 03]. The confidence measures introduced there consist of a combination of different features in a neural network. The confidence is estimated for a sequence of words in an interactive machine translation environment. The probability of being a correct extension of a correct sentence prefix is computed for this word sequence.

In this thesis, the application of confidence measures for translation prediction in an interactive SMT system will be investigated as well. The main differences to [Gandrabur and Foster 03] are the following:

- The SMT system applied in [Gandrabur and Foster 03] is a rather simple

model that combines a trigram language model and the IBM translation model 2 [Brown et al. 93]. In contrast to this, a fully-fledged SMT system on the basis of bilingual phrases [Och and Ney 04, Bender et al. 04] is applied in this thesis, which takes a large number of different sub-models into account.

- The system described in [Gandrabur and Foster 03] predicts extensions of up to four words whereas the RWTH system translates the whole input sentence.

- In addition to the prediction of words based on confidence, a new approach of rejecting words with low confidence is studied in this thesis.

- Furthermore, a method to make use of a given prefix (which is known to be correct) in the confidence estimation will be introduced here.

- In [Gandrabur and Foster 03], the confidence is estimated for the whole sequence of words which is a possible prediction. Unlike this, the confidence measures presented here operate on the word level.

In 2003, a team in the yearly summer workshop at the Center for Language and Speech Processing (CLSP) at Johns Hopkins University, Baltimore, MD, developed confidence measures for machine translation. The team was headed by the authors of [Gandrabur and Foster 03], George Foster and Simona Gandrabur. The team investigated the combination of several confidence features using neural networks and a naive Bayes classifier. The features applied there include those which had been developed by team members before [Gandrabur and Foster 03, Ueffing et al. 03]. Among them are also some of the word posterior probabilities which will be presented in this thesis.

The workshop team studied confidence estimation on the sentence level as well as on the word level, where the focus lay on the sentence level. A large variety of features was investigated which include heuristic and semantic features as well as word posterior probabilities. Many of them were calculated over the N-best lists generated by the ISI alignment templates system. The features can be grouped into five different classes:

- features depending on the SMT system which generated the translations, such as the probabilities assigned by the translation sub-models or search statistics from the translation process,

- target language based features, such as language model scores or semantic features extracted from WordNet,

- source sentence based features that try to capture the inherent translation difficulty of the input sentence, such as sentence length and language model scores,

- N-best list based features such as the system-based word posterior probabilities which will be presented in section 3.2,

- features that model the correspondence between source and target sentence, such as the feature based on IBM model 1 (see section 3.3.1).

Following the work of the summer workshop team, [Quirk 04] presents an investigation on different approaches to sentence-level confidence estimation. A number of features is computed for each sentence generated by an MT system and those features are combined using several different methods: modified linear regression, neural nets, support vector machines, and decision trees. Many of the applied sentence features are similar to those presented in [Blatz et al. 03]; the others are specific to the underlying MT system that generated the translations [Menezes and Richardson 01]. [Quirk 04] work also investigates the use of manually tagged data for training the confidence measures. The author found that using a small amount of manually labeled training data yields better performance than using large quantities of automatically labeled data.

1.2.2 Application of confidence measures

Apart from the use in interactive SMT systems as described above, several authors have applied confidence measures to improve MT quality.

[Blatz et al. 03] contains a short excursion on application of sentence-level confidence measures to rescoring on N-best lists and selection of output from two different SMT systems, but due to time restrictions in the CLSP summer workshop, no systematic experiments were conducted. Therefore, the results presented there are not too conclusive.

[Akiba et al. 04b] reports the application of confidence measures to the selection of output on N-best lists produced by different MT systems. Word-level confidence measures, namely the rank-weighted sum developed in this thesis (see chapter 3), are used to discard low-quality system output before selecting a translation among different MT systems.

[Jayaraman and Lavie 05] deals with the combination of output from different MT systems. The different hypotheses are combined on the word level to generate new translations. Since the original MT systems used in this work are non-statistical, a score for each word has to be calculated. This score is determined by a language model and several confidence scores which reflect the translation quality of the underlying MT system.

1.3 Bayes decision rule and word posterior probabilities

To the best of our knowledge, the "standard" version of the Bayes decision rule, which minimizes the number of sentence errors (see equation 1.1), is used in virtually all approaches to statistical machine translation (SMT). There are only a few research groups that do not take this type of decision rule for granted. In [Kumar and Byrne 04], an approach to SMT is presented that minimizes the posterior risk for different error measures. Rescoring is performed on 1,000-best lists produced by an SMT system. In [Och 03], a sort of error related or discriminative training is used, but the decision rule as such is not affected. In other research areas, e.g. in speech recognition, there exist a few publications that consider the word error instead of the sentence error for taking decisions [Goel and Byrne 03]. In this thesis, the loss functions for several word error

measures for MT will be formulated. This will show a close relation between the word posterior probabilities developed here and the posterior risk.

2 Scientific goals

In this chapter, conclusions are drawn from the state-of-the-art of confidence estimation and its applications. As discussed in the previous chapter, confidence measures for machine translation have hardly been investigated before. The goal of this thesis is to set up a probabilistic framework for the computation of word posterior probabilities for machine translation. This includes defining different concepts of word posterior probabilities as well as different ways of calculating them. The direct use of word posterior probabilities as confidence measures will be studied. Moreover, the application of confidence measures in different scenarios, namely rescoring and interactive machine translation, is to be investigated. In particular, the following aspects are considered in detail:

Word posterior probabilities

The main problem which has to be solved for the computation of word posterior probabilities is to define under which condition a word occurs in a sentence. Several solutions to this problem will be presented in this thesis. For example, the definition can be based on the target position of the word or its frequency in the sentence.

Different approaches to the calculation of word posterior probabilities will be introduced in chapter 3. They can be divided into two categories: Firstly, there are methods which derive the word posterior probability from the sentence posterior. To this end, they make use of system output, such as word graphs or N-best lists, which represent the space of possible target sentences. The word posterior probability can be determined by summing the probabilities of the sentences in this space which contain the target word. These approaches require information from the underlying statistical machine translation system which generated the translations. Secondly, direct methods will be developed which take other knowledge sources, such as word or phrase lexicon models, into account. They are independent of the translation system which generated the translation, and can thus be applied to non-statistical machine translation systems as well. These methods are based e.g. on HMMs or phrase-based translation models.

Various aspects of the computation of word posterior probabilities are studied in this thesis. In particular, these are the scaling of the probability density functions used in the underlying translation system, the optimization of the models w.r.t. confidence estimation, and normalization issues. Furthermore, an efficient implementation of the forward-backward algorithm on word graphs is presented.

Confidence measures

The word posterior probabilities proposed in this thesis can be directly used as word-level confidence measures. Different aspects of their use for classification of words as correct or incorrect will be discussed and evaluated. The first problem to be solved in this setting is how to determine the correctness of words. Normally, several correct ways of translating a sentence exist. As a consequence of this, evaluation for MT is still an unsolved problem. In this thesis, several widely used evaluation measures for MT will be considered and different ways of determining the correctness of single words will be derived from them.

The studied aspects of confidence measures further include: the definition of the acceptance threshold, and the relation between the confidence measure and the word error measure which defines the reference tags. All experiments will be evaluated with two different, well-established metrics. To achieve a systematic analysis, all confidence measures in this thesis are evaluated on various translation tasks and different language pairs.

The focus of this thesis is on the use of theoretically well-defined word posterior probabilities rather than combination of many heuristics. Nevertheless, feature combination will be discussed and experimental results will be given.

Improving translation quality through confidence estimation

In previous work, confidence measures have mostly been applied in relatively simple tasks. In this thesis, approaches improving the quality of state-of-the-art translation systems will be presented. It will be shown how confidence measures can improve interactive statistical machine translation where the underlying system is a cutting-edge phrase-based system. The confidence is used as a feature for the selection of words; and words with low confidence are rejected. Additionally, rescoring with confidence measures will be shown to improve translation quality. The statistical machine translation system which serves as baseline was ranked first in the international TC-STAR evaluation campaign in March 2005.

Connection with Bayes decision rule

In this thesis, the loss functions for several MT error measures will be formulated, and the posterior risk will be derived. It will be shown that this risk includes terms which are identical or very close to the word posterior probabilities used as confidence measures. This sets up a theoretical foundation for the different ways of defining word posterior probabilities. Naturally, these confidence measures can be expected to perform very well if the reference tags are defined by the corresponding word error measure.

Structure of this document

The organization of this thesis is as follows: First, the concept of word posterior probabilities will be presented in chapter 3. Several approaches to the definition of word posterior probabilities and details of their calculation will be explained. Chapter 4 will show how the word posterior probabilities introduced in chapter 3 can be used as confidence measures. Different ways of defining the true classes of single words in translation output will be presented. The evaluation measures used to assess the quality of confidence measures will be discussed. Furthermore, some considerations on estimating the acceptance threshold for the word posterior probabilities will be given. In chapter 5, the relation between Bayes decision rule for machine translation and word posterior probabilities will be investigated. The application of confidence measures in an interactive statistical machine translation system will be depicted in chapter 6. First, the idea of interactive statistical machine translation will be reviewed. Then, different methods will be explained which use confidence measures to improve the quality of the interactive system. The experimental results for all methods described in this thesis will be presented in chapter 7. First, the classification performance of the different confidence measures will be analyzed in detail. A comparison of the different approaches will be given on four different language pairs and different domains. Additionally, experimental results for rescoring and the application of confidence measures in interactive statistical machine translation will be presented. Chapter 8 will summarize the scientific contributions of this thesis, followed by an outlook in chapter 9.

The appendix contains details on the experimental settings and some additional experimental results and data analyses. Appendix A will present the corpora on which the experiments have been performed. The evaluation metrics for machine translation will be summarized in appendix B. Appendix C gives an overview of the machine translation systems used for the experiments reported in this thesis. The additional results of experiments and data analyses will be shown in appendix D. The symbols and acronyms will be explained in appendix E.

3 Word posterior probabilities

This chapter will present word posterior probabilities which are the basis for all approaches ot confidence estimation investigated in this thesis. In section 3.1, the concept of word posterior probabilities will be explained. Several issues of their computation will be discussed in general. Two types of word posterior probabilities will be shortly introduced: The first type are probabilities determined based on output of the MT system which generated the translation. These *system-based* word posterior probabilities make use of the underlying SMT models. The second type are the *direct* models. They use knowledge sources such as statistical word or phrase lexica and are independent of the underlying SMT system. Section 3.2 will describe several different concepts of system-based word posterior probabilities in detail. Their calculation over N-best lists and word graphs will be discussed. The direct models will be depicted in section 3.3. Several approaches will be presented: models based on the IBM translation model 1, on HMMs, and on a phrase-based translation model.

3.1 Introduction

Word posterior probabilities are the basis for all approaches to confidence estimation presented in this thesis. It will be explained in the following how they can be derived from the sentence posterior. The posterior probability of a sentence e_1^I can be obtained from the joint probability $p(e_1^I, f_1^J)$ which the statistical machine translation system assigns to a generated translation. The normalization of these probabilities by the term $p(f_1^J)$ results in a probability distribution over all target sentences, given the source sentence. As seen in section 1.1, the sentence probabilities employed in search are not normalized, which does not affect the result of the search. But for the use in confidence estimation, they need to be normalized in order to obtain a probability distribution over all target sentences. It will be explained later in this section how to perform this normalization (see equation 3.3). From the sentence posterior probabilities, the word posterior probabilities can be calculated by summing up the probabilities of all sentences containing the target word. Nevertheless, it is not clear which criterion should be applied to decide whether two sentences contain the same word under the same condition or not. The answer to this question is not at all trivial. Due to ambiguities, the word position in the sentence is not fixed. Sentences can have different numbers of words because of deletions and insertions. Additionally, the words can be reordered in different ways during the translation process. The posterior probability of a target word e can depend on the occurrence in position i of the target sentence, for example, or on the occurrence in any position of the target sentence or on the number of times the word is contained in the sentence. Thus, several different criteria for comparing sentences are defined and investigated in this thesis. The

basic concept of calculating the posterior probability will be explained for the target word e occurring in position i of the sentence. This is a rather strict and simple criterion. Section 3.2 will describe several different concepts of word posterior probabilities which relax this strict assumption.

Let $p(e_1^I, f_1^J)$ be the joint probability of source sentence f_1^J and target sentence e_1^I. The word posterior probability of e occurring in position i is calculated as the normalized sum of probabilities of all sentences containing e in exactly this position:

$$p_i(e \mid f_1^J) \;=\; \frac{p_i(e, f_1^J)}{\sum\limits_{e'} p_i(e', f_1^J)} \;, \tag{3.1}$$

where

$$p_i(e, f_1^J) \;=\; \sum_{I, e_1^I} \delta(e_i, e) \cdot p(f_1^J, e_1^I) \;. \tag{3.2}$$

Here $\delta(\cdot, \cdot)$ is the Kronecker function. The normalization term in equation 3.1 is

$$\sum_{e'} p_i(e', f_1^J) \;=\; \sum_{I, e_1^I} p(f_1^J, e_1^I) \;=\; p(f_1^J) \;. \tag{3.3}$$

This raises the question of how to calculate the sums over the possible target sentences as given in equations 3.2 and 3.3. In this work, two different ways have been investigated: The translation hypotheses space can be approximated via a word graph or an N-best list. The summation is then performed explicitly over all sentences given in this restricted space. In the case of the N-best list, this is straightforward because the sentences are already listed. On the word graph, the forward-backward algorithm can be applied to efficiently carry out the summation. This approach will be called **system-based** in the following, because the calculation depends on the output of the SMT system which generated the translations. The summation space is restricted to those hypotheses which are assigned a high probability by the SMT system, and the others are not considered. The sentence probabilities summed in equation 3.2 are the scores given by the base SMT system.

The second approach pursued here is the summation using **direct** models such as IBM model 1, HMMs, or a phrase-based model. The HMM based approach does not explicitly list the possible target sentences in the summation. Instead, forward-backward calculation is applied to determine the word posterior probabilities. The methods based on IBM model 1 and the phrase-based translation model do not consider the whole target sentence at all. The sum over single words or phrases without context. These model based word posterior probabilities are **independent** from the system generating the translations. They do not require the MT system to assign a probability to the translation hypothesis. Thus, they can also be used for confidence estimation if the hypotheses come from a non-statistical MT system or if only the single best translations without any scores are given. The word posterior probabilities based on direct models have been tested on various systems in this work. Among them is also a non-statistical MT system by Systran [sys 05].

Word graph based calculation

A word graph represents the most promising hypotheses generated by the translation system [Ueffing et al. 02, Zens and Ney 05]. This has the advantage of being a compact representation of the search space which allows for efficient calculation of word posterior probabilities. On the other hand, some of the word posterior probabilities described in the following, such as the one based on word counts, cannot be efficiently determined over the word graph. Details on the word graph based calculation of word posterior probabilities will be given in the following subsections. Here, the general concept will be shortly presented.

A word graph is a directed acyclic graph $G = (V, E)$ with vertex set V and edge set E. It has one designated root node $n_0 \in V$, representing the beginning of the sentence. Each path through the word graph represents a translation candidate. The nodes of the graph contain information such as the set of covered source positions and the language model history. Two hypotheses can be recombined if their information is identical. The edges are annotated with target words. Additionally, they contain scores representing the part of the probability that is assigned to this word as part of the target hypothesis. When multiplying the scores along a path, the probability of the corresponding hypothesis is obtained. For an example of a word graph, see figure 3.1 on page 19. The source sentence is "wir können das machen!", and the reference translation is "we can do that!". The leftmost node is the root node n_0. The other nodes represent different states w.r.t. the set of covered source positions and language model history. In this example, a trigram language model is applied, i.e. all paths leading into a node share the last two words. The translation alternatives contained in this word graph represent different reorderings of the words in the sentence: the monotone translation "that do" as well as the correctly reordered sequence "do that" occur. Note that in order to limit the size of the graph and keep the presentation simple, an example has been chosen where all sentences have the same length.

The posterior probabilities of words can be computed by summing up the probabilities of all paths in the graph which contain an edge annotated with the corresponding word in the target position under consideration. This summation is performed efficiently using the forward-backward algorithm. The algorithm also determines the total probability mass which is needed for normalization as shown in equation 3.3.

N-best list based calculation

An N-best list contains the N most promising translation hypotheses generated by the statistical machine translation system. They are sorted by their probability in descending order. The N-best list is extracted from the word graph. This representation allows for easy computation of the sum given in equation 3.2. So, the calculation of more complex variants of word posterior probabilities, such as the count-based approach (see section. 3.2.3), is feasible.

The word posterior probabilities presented in equation 3.1 are calculated by summing the sentence probabilities of all sentences containing target word e in target position

i. The sentence probability $p(f_1^J, e_{n1}^{nI_n})$ is given in the N-best list. Instead of the sum of probabilities, one can also determine the relative frequency or the rank-weighted frequency of a word as follows: The relative frequency of e occurring in target position i in the N-best list is computed as

$$h_i(e \mid f_1^J) \;:=\; \frac{1}{N} \sum_{n=1}^{N} \delta(e_{ni}, e) \;. \tag{3.4}$$

The rank-weighted frequency is determined as

$$r_i(e \mid f_1^J) \;:=\; \frac{2}{N(N+1)} \sum_{n=1}^{N} \delta(e_{ni}, e) \cdot (N+1-n) \;. \tag{3.5}$$

Here, the inverted ranks $N+1-n$ are summed up because an occurrence of the word in a hypothesis near to the top of the list shall score better than one in the lower ranks. This value is normalized by the sum of all ranks in the list. Note that the values in equations 3.4 and 3.5 could also be calculated over N-best lists which do not contain the sentence probability. Moreover, they can be applied in situations where the hypotheses in the N-best list differ only slightly from each other. In this situation, the sentence probabilities will be very close to each other. The rank-weighted summation provides a method which distinguishes more distinctly between different translation alternatives by exploring the ranking information given in the N-best list.

When calculating word posterior probabilities over N-best lists, the normalization term used in equation 3.3 equals the probability mass contained in this N-best list. Thus, it can be determined very efficiently by summing the sentence probabilities of all sentences $e_{n1}^{nI_n}$ in the N-best list.

Direct models

Instead of approximating the summation space through word graphs or N-best lists, one can use direct models such as the IBM model 1 or HMMs for estimating word posterior probabilities. This allows for an exact summation in equation 3.2, but it has the disadvantage that the underlying models are weaker.

The IBM model 1 based calculation of word posterior probabilities, as described in section 3.3.1, does not explicitly list the possible target sentences and considers only the target word e without any context. It does not take the position of the word in the sentence into account which leads to a very simple formula for the word posterior probability.

The HMM (see section 3.3.2) on the other hand integrates the context of each word, because it is a first order model. Moreover, it explicitly considers the exact position of the word in the sentence. The summation over all possible target sentences can be performed via the forward-backward algorithm.

The direct phrase-based model that will be presented in section 3.3.3.2 sums the probabilities of all bilingual phrases $(\tilde{f}, \tilde{e})$ where $\tilde{f}$ is part of the source sentence f_1^J, and the target phrase $\tilde{e}$ contains the word e. Thus, it considers some context of the target

word, but it completely disregards the context of the phrase $\tilde{e}$. That is, the actual target sentences are not considered, but only the parts matching e. This approach does not take the target position of the word into account. It is based on a state-of-the-art translation model and has the advantage of being a model which can be easily calculated.

Scaling factors

During the translation process, the different sub-models (such as the language model and the lexicon model) are weighted differently. These weights or scaling factors can be optimized w.r.t. some evaluation criterion [Och 03]. A more detailed description of this will be given in section 3.3.3.1. Nevertheless, this optimization determines only the relation between the different models, and not the absolute values of the scaling factors. The absolute values are not needed for the translation process, because the search is performed using the maximum approximation. In contrast to this, the actual values of the weights make a difference for confidence estimation, because the summation over the sentence probabilities is performed. To account for this and find the optimal values of the scaling factors, a global weight λ is introduced which scales the sentence probability. The word posterior probability is then calculated according to

$$p_i(e \mid f_1^J) = \frac{\sum\limits_{I,e_1^I} \delta(e_i, e) \cdot p^\lambda(f_1^J, e_1^I)}{\sum\limits_{e'} \sum\limits_{I,e_1^I} \delta(e_i, e') \cdot p^\lambda(f_1^J, e_1^I)} \; . \tag{3.6}$$

When determining word posterior probabilities over an N-best list, for example, this scaling factor is optimized on a development set distinct from the test set.

In the direct phrase-based models, it is possible to optimize the weights of the different sub-models immediately w.r.t. classification performance, measured by one of the evaluation metrics described in section 4.3. This issue will be discussed in further detail in section 3.3.3.2.

3.2 System-based concepts of word posterior probabilities

In this thesis, different concepts of word posterior probabilities have been investigated which will be described in the following. In chapter 5, it will be shown how they are related to Bayes decision rule for different error measures for MT. It is not intuitively clear how to define the condition under which a sentence contains a specific word: the word e can be considered in its exact target position i, or in a window $i \pm t,\ t \in \mathbb{N}$, around this position. Alternatively, the number of occurrences of the word can be taken into account or the surrounding words could serve as indicators. This work presents the approaches which proved most promising from a theoretical viewpoint and in the experimental evaluation.

Table 3.1 gives an example illustrating several different variants of word posterior probabilities which will be explained in this subsection. Assume that the four sentences composed of the words 'A' to 'E' represent the space of all possible translations. One

Table 3.1: Illustration of different concepts of word posterior probabilities: Sentences taken into account for the calculation of the word posterior probability of the word 'A' in target position 4 are marked with a '+'.

sentence	fixed (eq. 3.2)	Levenshtein (eq. 3.8)	window ±1 (eq. 3.9)	count (eq. 3.13)	context ±1 (eq. 3.14)
A B C **A** D E	+	+	+	+	+
B B E A D	+	+	+		
A B A D E		+	+	+	
A A D E		+		+	

can see that the positions of the words in the target sentence differ due to insertions, deletions and reordering. The confidence of word 'A' in position 4 in the first sentence is to be calculated. The table shows which sentences will be considered for the summation in equation 3.2 if the different concepts of word posterior probabilities are applied: The approach considering the "fixed" target position (see section 3.2.1.1) takes only the first two sentences into account, whereas the approach based on Levenshtein-alignment (see section 3.2.1.2) and the windowing (see section 3.2.1.3) yield a more reliable confidence estimation by counting more sentences. The count-based method (see section 3.2.3) sums over all sentences containing 'A' at least twice. The strictest concept here is the one looking at the context of the word (see section 3.2.4): only the first translation meets this criterion. This is why the resulting word posterior probability is by far the lowest.

3.2.1 Approaches based on target position

As mentioned above, the concept of word position is not unambiguous when comparing different translation hypotheses. Therefore, different ways of regarding the word position are investigated in this work. Apart from considering the *fixed* target position i, the concept of position can also be relaxed by introducing a window over the surrounding target positions or by performing a Levenshtein alignment between the different translation alternatives. These different concepts and efficient methods for calculating the word posterior probabilities will be presented in the following.

3.2.1.1 Fixed target position

In this approach, the word posterior probability is determined for word e occurring in target position i as shown in equation 3.2. This variant requires the word to occur exactly in the given position i. Here, a probability distribution over the pairs (e, i) of target words e and positions i in the target sentence is obtained. This type of word posterior probability is related to the error measure Pos which will be explained in section 5.3. The concept of word posterior probabilities based on the fixed target position allows for easy calculation over word graphs and N-best lists. It is also the basis for the HMM approach described in section 3.3.2. However, this concept is rather restrictive. In practice, the target position of a word varies between different translation alternatives. The method presented here is

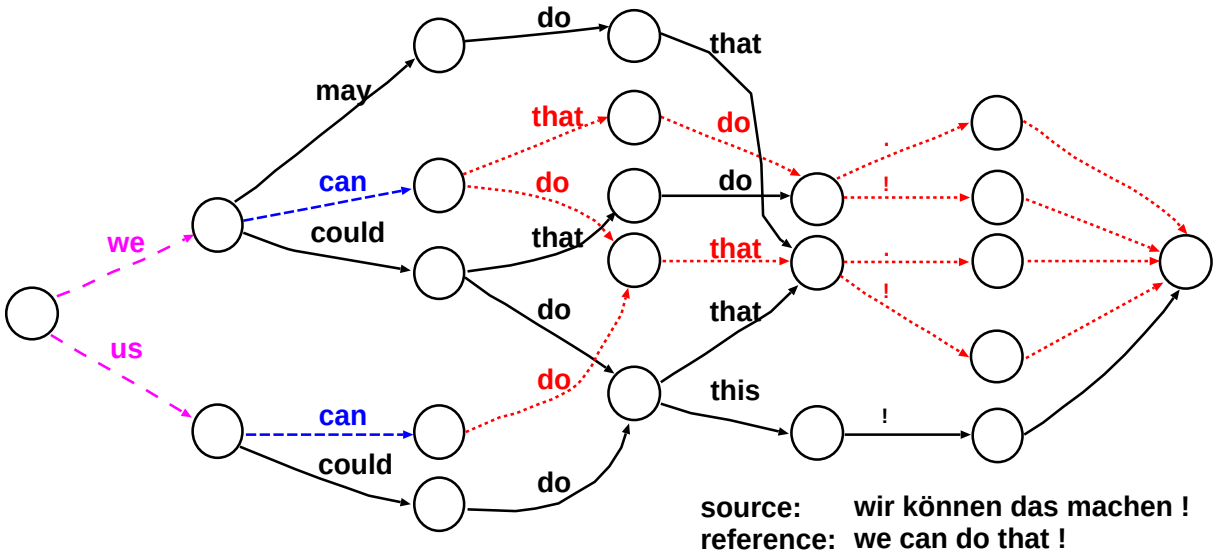


Figure 3.1: Example of forward-backward calculation on a word graph.

a starting point for more flexible approaches which perform summation over a window of target positions.

Implementation using N-best lists

The calculation of the word posterior probabilities given in equation 3.2 over an N-best list is straightforward: For each sentence contained in the list, the word in position i is compared to e. If they match, the sentence probability is added to the sum. This sum is then normalized by the total probability mass contained in the list, as described in equation 3.3.

Implementation using word graphs

The computation of word posterior probabilities on word graphs can be performed by applying a **forward-backward algorithm**. Let n' and n be nodes in a word graph, and (n', n) the directed edge connecting them. The edge is annotated with a target word which is denoted by $e((n', n))$ and a probability, denoted by $p((n', n))$. This is the probability which this word contributes to the overall sentence probability. The word posterior probability is obtained by summing up the path probabilities of the edges which are annotated with the corresponding word. The notion of $p((n', n))$ given here is rather imprecise and will be used only to explain the basic concept of the algorithm. Afterwards, the exact equations for an IBM model 4 translation model will be given. The basic idea of the forward-backward algorithm is to split the sum over all sentence probabilities into two parts which can be recursively calculated. The **forward probability** is the sum over all paths up to position i, and the **backward probability** is summed over all outgoing paths. These probabilities are then combined to obtain the overall word posterior probability.

The forward probability $\Phi_i((n',n))$ of an edge (n',n) is the probability of reaching this edge from the source of the graph, where the word $e((n',n))$ is the i-th word on the path. It can be obtained by summing the probabilities of all incoming paths of length $i-1$, which allows for recursive calculation. This leads to the following formula:

$$\Phi_i((n',n)) \;=\; p((n',n)) \cdot \sum_{n''} \Phi_{i-1}((n'',n')) \;.$$

The backward probability $\Psi_i((n',n))$ expresses the probability of completing a sentence from the current edge onwards, i.e. of reaching the sink of the graph. It can be recursively determined in descending order of i as follows

$$\Psi_i((n',n)) \;=\; p((n',n)) \cdot \sum_{n^*} \Psi_{i+1}((n,n^*)) \;.$$

Then, the forward and backward probabilities of all edges which are annotated with e are combined. That is, the total path probabilities of all hypotheses containing target word e in position i are summed up. This yields

$$p_i(e, f_1^J) = \sum_{(n',n)} \delta(e((n',n)), e) \cdot \frac{\Phi_i((n',n)) \cdot \Psi_i((n',n))}{p((n',n))} \;,$$

which is normalized to obtain word posterior probabilities:

$$p_i(e \,|\, f_1^J) \;=\; \frac{p_i(e, f_1^J)}{\sum_{e'} p_i(e', f_1^J)} \;. \tag{3.7}$$

Note that (for computational reasons) the term $p((n',n))$ is included both in the forward and in the backward probability so that the product has to be divided by this term.

The normalization term given in equation 3.7 corresponds to the probability mass α contained in the word graph. This can be calculated in two ways: Either as the sum of backward probabilities of all edges leaving the source node n_0. Or as the sum of forward probabilities of all incoming edges of the sink s of the graph. That is,

$$\begin{aligned} \alpha \;&=\; \sum_{(n_0,n)} \Psi_1(n_0,n) \\ &=\; \sum_I \sum_{(n,s)} \Phi_I(n,s) \;, \end{aligned}$$

where I is the last position in the generated target sentences.

Figure 3.1 illustrates the forward-backward algorithm. Assume the word posterior probability of word “can” appearing in the second position of the target sentence is to be calculated. There are two edges in the graph which contain this word in the desired target position; they are marked by dashed blue lines. Thus, the probabilities of the paths leading through these edges have to be summed. The forward probabilities are the probabilities of the incoming edges, shown in dashed magenta. The backward probabilities are those of the paths marked in dotted red. They are combined (separately for each edge) and then added up to the word posterior probability of “can”.

Exact equations for the IBM translation model 4

In the following, the exact equations of the forward-backward algorithm for the IBM translation model 4 will be given. Assume a word graph has been constructed using IBM model 4 so that the nodes contain the following information:

- the set of covered source sentence positions for the partial hypothesis,
- the source position which was covered last.

The edges are annotated with

- the generated target word,
- the probabilities of the different IBM-4 sub-models for this word: fertility, distortion, and lexicon model.

The language model probabilities do not have to be contained in the graph, because they can be calculated on the fly. Thus, the language model history does not have to be stored in the nodes. Consequently, more partial hypotheses can be recombined which reduces the density of the word graph drastically.

Consider e occurring in position i of the target sentence e_1^I, so that $e_i = e$. Let B_i be the set of source positions aligned to target position i. The information relevant for determining the word probabilities of a word e_i in position i is the following:

- the position i of the word in the target hypothesis,
- the translation probability $p(f_{B_i}|e_i)$, where f_{B_i} is the set of source words aligned to e_i; this probability includes lexicon and fertility model probability,
- the alignment probability $p(B_i|B_{i-1})$, where B_{i-1} is the set of source positions covered by the predecessor target word e_{i-1}.

Assume a trigram language model. The formulae which will be presented here can be extended to capture longer histories as well, but in order to keep the presentation understandable, trigrams will be used.

In the following, let C_i be the set of source positions which are covered by the partial hypothesis $e_1 \ldots e_i$, i.e. $C_i := \bigcup_{k=1}^{i} B_k$. The information about this coverage set C_i is stored in the nodes of the word graph. Throughout the presentation, the complement of a set C is denoted by $\overline{C}$, and the difference of two sets B and C by $B \setminus C$.

The forward probability of word e_i occurring in position i depends on the predecessor word e_{i-1}, its coverage set, and the set B_i. It can be computed as follows:

$$\begin{aligned}&\Phi_i(e_{i-1}, C_{i-1}; e_i, B_i) = \\ &= p(f_{B_i}|e_i) \cdot \sum_{e_{i-2}} \sum_{B_{i-1} \subseteq C_{i-1}} \Big[p(B_i|B_{i-1}) \cdot p(e_i|e_{i-1}e_{i-2}) \cdot \Phi_{i-1}(e_{i-2}, C_{i-1} \setminus B_{i-1}; e_{i-1}, B_{i-1}) \Big] ,\end{aligned}$$

which is calculated recursively in ascending order of i. The backward probability, i.e. the probability of completing a sentence from word e_i in position i is

$$\begin{aligned}&\Psi_i(e_i, B_i; e_{i+1}, \overline{C_i}) = \\ &= p(f_{B_i}|e_i) \cdot \sum_{e_{i+2}} \sum_{B_{i+1} \subseteq \overline{C_i}} \left[p(B_{i+1}|B_i) \cdot p(e_{i+2}|e_{i+1}e_i) \cdot \Psi_{i+1}(e_{i+1}, B_{i+1}; e_{i+2}, \overline{C_{i+1}}) \right] .\end{aligned}$$

This can be determined recursively in descending order of i. Combining these probabilities using the forward-backward algorithm leads to

$$\begin{aligned}&p_i(e, f_1^J) = \\ &= \sum_{e_{i-1}} \sum_{e_{i+1}} \sum_{C_{i-1}} \sum_{B_i} p(e_{i+1}|e_i e_{i-1}) \cdot \frac{\Phi_i(e_{i-1}, C_{i-1}; e_i, B_i) \cdot \Psi_i(e_i, B_i; e_{i+1}, \overline{C_i})}{p(f_{B_i}|e)} .\end{aligned}$$

Note that $C_i = C_{i-1} \cup B_i$ so that $\overline{C_i}$ is known for the calculation of the backward probability.

As stated before in this section, the normalization can be performed by dividing the term $p_i(e, f_1^J)$ by the total probability mass α contained in the word graph. This value can be calculated as

$$\begin{aligned}\alpha &= \sum_{e_1} \sum_{B_1} \sum_{e_2} \Psi_1(e_1, B_1; e_2, \overline{B_1}) \cdot p(e_2|e_1) \cdot p(e_1) \\ &= \sum_{I} \sum_{e_I} \sum_{B_I} \sum_{e_{I-1}} \Phi_I(e_{I-1}, \overline{B_I}; e_I, B_I) .\end{aligned}$$

Here, I is the last position in the generated target sentences. These sums correspond to the sum of the backward probabilities of all paths leaving the root node and to the sum of forward probabilities of all paths reaching the sink of the graph.

3.2.1.2 Levenshtein-aligned target position

In different translations of a source sentence, the same target word can occur in different positions. One way of accounting for this when calculating word posterior probabilities is to perform the Levenshtein alignment [Levenshtein 66] between sentence e_1^I under consideration and the other target sentences. The summation in equation 3.2 is then performed over all sentences containing e in position i or in a position Levenshtein-aligned to i. Let $\mathcal{L}(e_1^I, \tilde{e}_1^{\tilde{I}})$ be the Levenshtein-alignment between sentences e_1^I and $\tilde{e}_1^{\tilde{I}}$, and $\mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}})$ that of word e in position i in e_1^I. Then, the word posterior probability of word e occurring in a position Levenshtein-aligned to i is given by

$$p_{i,Lev}(e \mid f_1^J, e_1^I, \mathcal{L}) = \frac{p_{i,Lev}(e, f_1^J, e_1^I, \mathcal{L})}{\sum_{e'} p_{i,Lev}(e', f_1^J, e_1^I, \mathcal{L})} , \tag{3.8}$$

where

$$p_{i,Lev}(e, f_1^J, e_1^I, \mathcal{L}) = \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} \delta(e, \mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}})) \cdot p(f_1^J, \tilde{e}_1^{\tilde{I}}) .$$

The probability depends on all target words in the hypothesis e_1^I under consideration, because the Levenshtein alignment $\mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}})$ is used. This concept of word posterior probabilities is inspired by the error measure WER (see section 4.2).

The implementation of the calculation over N-best lists is straightforward: The Levenshtein alignment is performed between the hypothesis e_1^I and every sentence contained in the N-best list individually, and then the summation is carried out as given in the equation above. For word graphs, no efficient way of determining the Levenshtein alignments and the resulting word posterior probabilities is known.

3.2.1.3 Window over target positions

Another way of accounting for slight variations in the target position i of word e is the introduction of a window $i \pm t,\ t \in \mathbb{N}$, around position i. The word confidence is determined as the sum of the word posterior probabilities calculated for the positions within this window. This leads to

$$p_{(i,t)}(e \,|\, f_1^J) \;=\; \sum_{k=i-t}^{i+t} p_k(e \,|\, f_1^J)\ . \tag{3.9}$$

The windowing can easily be integrated both into the N-best list and the word graph based implementation: the target position dependent word posterior probabilities are calculated as stated in equation 3.2, and the summation over the positions in the window can be performed in an additional step.

3.2.1.4 Average over all target positions

When considering error measures such as PER for evaluating MT, the word order in the target sentence is not taken into account. A similar idea can be pursued in confidence estimation by averaging the position-based word posterior probabilities for a target word e over *all* positions in the sentence. This corresponds to averaging over a target window $i \pm t$ covering the whole sentence. The word posterior probability is then calculated as

$$p_{avg}(e \,|\, I, f_1^J) \;=\; \frac{1}{I} \sum_{k=1}^{I} p_k(e \,|\, f_1^J)\ . \tag{3.10}$$

Analogously to the windowing approach presented above, the averaging can be performed after calculating the position dependent word posterior probability $p_i(e \,|\, f_1^J)$ for each target position i. Note that the target sentence length I is fixed.

3.2.1.5 Any target position

In this definition of word posterior probabilities, all sentences are taken into account that contain the target word e. The position is completely disregarded. This word posterior probability does not depend on the actual word sequence, but only on the so-called "bag of

words" which regards the identities of the words, but not their order. It can be motivated by the error measure 'Set' presented in section 4.2. The word confidence is calculated as

$$p_{any}(e, f_1^J) = \sum_{i=1}^{I_{\max}} \sum_{I \geq i} \sum_{e_1^I} \delta(e_i, e) \cdot p(f_1^J, e_1^I) \ , \tag{3.11}$$

which has again to be properly normalized. $I_{\max}$ is the maximal target sentence length. This concept can easily be realized when working with N-best lists, but the implementation using word graphs would require keeping track of all words contained on a path. In the work presented here, only the N-best list based implementation has been investigated.

The confidence measure based on IBM model 1 which will be described in section 3.3.1 pursues a "bag of words" approach as well, but without explicitly listing all possible target sentences.

3.2.2 Approach based on source position

Another way of accounting for the occurrence of word e in different positions of the target sentence is to consider the source word(s) generating e. This type of word posterior probability was investigated in the experiments conducted during the CLSP summer workshop on confidence estimation for machine translation [Blatz et al. 03]. The provided N-best lists contain this information (unlike the ones generated by the current RWTH translation systems). This approach completely disregards the target position in which e occurs, and instead calculates the word posterior probability of e as a translation of the source word(s) aligned to it. Let B be the set of source positions aligned to target word e and B_1^I the sequence of these sets for the target sentence e_1^I. Then the word posterior probability is given by

$$p_B(e \,|\, f_1^J) = \frac{p_B(e, f_1^J)}{\sum_{e'} p_B(e', f_1^J)} \ ,$$

where

$$p_B(e, f_1^J) = \sum_{I, (e_1^I, B_1^I)} \delta((e, B) \in (e_1^I, B_1^I)) \cdot p(f_1^J, e_1^I, B_1^I) \ . \tag{3.12}$$

Here $\delta(\cdot)$ denotes an extension of the Kronecker delta:

$$\delta(a) = \begin{cases} 1 & \text{if a is true} \\ 0 & \text{otherwise} \end{cases}$$

If the information on the source positions covered by each target word is contained in the word graph or N-best list, the implementation is analogous to the one described in section 3.2.1.1. It has to be modified to account for the set B of covered source positions instead of the target position i. The coverage sets of the words on the forward/backward paths have to be stored. The general structure of the forward-backward algorithm is identical.

3.2.3 Count-based approach

Inspired by the posterior risk for PER (see equation 5.3 in section 5), the word posterior probability can be defined by taking the counts of the words in the generated sentence into account. The probability of target word e occurring in the sentence n times is determined as

$$p_e(n|f_1^J) = \frac{p_e(n, f_1^J)}{\sum_{n'=0}^{N_{\max}} p_e(n', f_1^J)} , \tag{3.13}$$

where

$$p_e(n, f_1^J) = \sum_{I, e_1^I} \delta(n_e, n) \cdot p(f_1^J, e_1^I) .$$

Here, n_e is the count of word e in sentence e_1^I, and $N_{\max}$ is the maximal count which is observed. This term does not depend on the actual word sequence, but only on the counts of the target words. Let n_1^E be the counts of all target words $1, \ldots, E$ in sentence e_1^I. In practice, many of these counts will be zero, of course. The posterior probabilities can then be expressed by the distribution over those count sequences:

$$p_e(n, f_1^J) = \sum_{n_1^E} \delta(n_e, n) \cdot p(n_1^E, f_1^J)$$

Using this concept, the target position of the word is not taken into account, but the first occurrence of a word in the sentence will obtain a word posterior probability different from that of the second occurrence. In the example presented in table 3.1, the first 'A' has a higher word posterior probability than the second one because all sentences in the list contain at least one 'A', but not all of them have two 'A's.

The summation in equation 3.13 can be performed over N-best lists (analogously to the word posterior probability variants described so far), but it cannot efficiently be determined over the word graph. The normalization term in equation 3.13 corresponds to the total probability mass contained in the N-best list, since also the case $n' = 0$ is included.

3.2.4 Context-based approach

Another possible definition of word posterior probabilities is to fix the context of word e, i.e. its predecessor e_- and/or its successor e_+. This gives an estimation of whether the target word is correct in this context. The exploration of context information is inspired by the MT evaluation metrics like BLEU which measure precision on the level of n-grams ($n = 1, \ldots, 4$). If only the predecessor is considered, the word posterior probability is given by

$$p_{e_-}(e \,|\, f_1^J) = \frac{p_{e_-}(e, f_1^J)}{\sum_{e'} p_{e_-}(e', f_1^J)} , \tag{3.14}$$

where

$$p_{e_-}(e, f_1^J) \;=\; \sum_{I,e_1^I} \delta((e_-, e) \subseteq e_1^I) \cdot p(f_1^J, e_1^I) \;.$$

The extension to the successor or the ± 1 context of the word is straightforward. This variant of word posterior probabilities has been implemented for N-best lists. Since it did not show good performance in the experiments, the approach has no longer been pursued, and the word graph based method has not been tested.

3.3 Direct models

In the following, the confidence measures based on direct models will be described. These approaches model the word posterior probability directly instead of summing the probabilities of sentences containing the target word. Confidence measures based on IBM model 1, HMMs, and phrase-based translation models have been developed and will be presented in this thesis.

3.3.1 Confidence measures based on IBM model 1

In [Brown et al. 93], five statistical models for machine translation were introduced. They are single word based translation models of increasing complexity. The so-called IBM model 1 (which is the simplest model) can be used in confidence estimation for determining the translation probability of a target word e averaged over the source sentence words [Blatz et al. 03]. The IBM model 1 uses what is known as a "bag of words" translation model, meaning that its calculations are not tied to any specific alignment structure (apart from the basic one-to-many source-target correspondence assumed by all IBM models). Rather, for each source/target hypothesis pair, the sum of probabilities of all possible alignments is determined. This captures a sort of topic or semantic coherence in translations. The word posterior probability is calculated according to the formula

$$p_{\text{ibm1}}(e \,|\, f_1^J) \;=\; \frac{1}{J+1} \sum_{j=0}^{J} p(e|f_j) \;. \tag{3.15}$$

Here, f_0 is the "empty" source word [Brown et al. 93]. The probabilities $p(e|f_j)$ are word-based lexicon probabilities, i.e. they express the probability that e is a translation of the source word f_j. The confidence measure was developed at the CLSP summer workshop 2003. Investigations on the use of the IBM model 1 for word-level confidence estimation showed promising results [Blatz et al. 03, Blatz et al. 04]. Thus, this method will be compared to other types of confidence measures here.

However, an analysis of the IBM-1 based confidence measure showed that the average over the lexicon probabilities is dominated by the maximum. Therefore, the following modification of the confidence measure presented in equation 3.15 is investigated here as well:

$$p_{\text{maxlex}}(e \,|\, f_1^J) \;=\; \max_{j=0,\ldots,J} p(e \,|\, f_j) \;. \tag{3.16}$$

This confidence measure considers only the maximum lexicon probability of target word e over the source sentence.

3.3.2 HMM based confidence measures

In this section, the use of HMMs for the calculation of word posterior probabilities will be explained. First, the HMM alignment model will be depicted in section 3.3.2.1. In section 3.3.2.2, the monotone model and its application in confidence estimation will be described. Section 3.3.2.3 will present the inverted HMM and the word posterior probabilities derived from it.

3.3.2.1 Hidden Markov models (HMMs)

As already stated in section 1.1, the correspondence between words of the source and the target language in the translation process is expressed by a so-called alignment model. Using this model, the probability $Pr(f_1^J, a_1^J \,|\, e_1^I)$ in equation 1.2 can be rewritten as

$$\begin{aligned} Pr(f_1^J, a_1^J \,|\, e_1^I) &= Pr(J \,|\, e_1^I) \cdot \sum_{j=1}^{J} Pr(f_j, a_j \,|\, f_1^{j-1}, a_1^{j-1}, e_1^I) \\ &= Pr(J \,|\, e_1^I) \cdot \sum_{j=1}^{J} Pr(a_j \,|\, f_1^{j-1}, a_1^{j-1}, e_1^I) \cdot Pr(f_j \,|\, f_1^{j-1}, a_1^j, e_1^I) \ . \end{aligned}$$

This decomposition results in three different probability models: the length model $Pr(J \,|\, e_1^I)$, the alignment model $Pr(a_j \,|\, f_1^{j-1}, a_1^{j-1}, e_1^I)$, and the lexicon model $Pr(f_j \,|\, f_1^{j-1}, a_1^j, e_1^I)$. The length model is normally omitted in praxis.

A first order alignment model based on **Hidden Markov Models (HMMs)** was proposed in [Vogel et al. 96]. In this model, the assumption is made that the alignment $a_j = i$ of a source position j depends only on its predecessor, $a_{j-1} = i'$, and the length I of the target sentence. The lexicon model is based only on single words without context. This yields

$$\begin{aligned} Pr(a_j \,|\, f_1^{j-1}, a_1^{j-1}, e_1^I) &= p(a_j \,|\, a_{j-1}, I) \\ Pr(f_j \,|\, f_1^{j-1}, a_1^j, e_1^I) &= p(f_j \,|\, e_{a_j}) \ . \end{aligned}$$

To make the alignment parameters independent of absolute word positions, it is assumed that the alignment probabilities depend only on the jump width $a_j - a_{j-1}$. This leads to a so-called **homogeneous** HMM with alignment probabilities of the form:

$$p(a_j \,|\, a_{j-1}, I) = \frac{c(a_j - a_{j-1})}{\sum_{a'_j=1}^{I} c(a'_j - a_{j-1})} \ ,$$

where $c(a_j - a_{j-1})$ is the count of jump width $a_j - a_{j-1}$ in the training corpus.

Originally proposed without an **empty word**, this model was extended in [Och and Ney 03]. The empty word serves as an artificial target word that generates the source words which do not have any direct correspondence in the target sentence. Each target word is assigned an additional empty word which is placed in position $i + I$. That is, a source word in position j is aligned to the empty word e_{i+I} if the previously visited target word is e_i (i.e. $a_{j-1} = i$). Since there is no jump width to be considered in this case, a probability p_0 for a transition to the empty word is introduced. This yields the following alignment probabilities ($i, i' \leq I$):

$$\begin{aligned} p(i + I \,|\, i', I) &= p_0 \cdot \delta(i, i') \\ p(i + I \,|\, i' + I, I) &= p_0 \cdot \delta(i, i') \\ p(i \,|\, i' + I, I) &= p(i \,|\, i', I) \end{aligned}$$

That is, alignments to target positions $i \leq I$ depend on the last target position $i' \leq I$ that corresponds to a real target word. All preceding empty word alignments are ignored.

Word posterior probabilities

As stated in section 3.1, the HMM model has the advantage that the sum over all possible target sentences can be carried out explicitly and is not approximated by a word graph or an N-best list. The HMM can be used to determine word posterior probabilities of the following form:

$$p_i(e \,|\, f_1^J, I) = \frac{p_i(e, f_1^J, I)}{\sum_{\tilde{e}} p_i(\tilde{e}, f_1^J, I)} ,$$

where

$$\begin{aligned} p_i(e, f_1^J, I) &= \sum_{e_1^I : e_i = e} p(e_1^I, f_1^J) \\ &= \sum_{e_1^I : e_i = e} p(e_1^I) \cdot \sum_{a_1^J} p(f_1^J, a_1^J \,|\, e_1^I) \\ &= \sum_{e_1^I : e_i = e} p(e_1^I) \cdot \sum_{a_1^J} \prod_{j=1}^{J} p(a_j \,|\, a_{j-1}, I) \cdot p(f_j \,|\, e_{a_j}) \ . \end{aligned} \tag{3.17}$$

Note that here the target sentence length I is kept fixed. It is the length of the target hypothesis for which the word confidences are to be calculated. This is a difference to the other confidence measures presented in this thesis: The word graph or N-best list based methods sum over target sentences of different lengths. The IBM model 1 and the direct phrase-based approach do not consider the target sentence length at all, because they do not include the context of the target word or phrase, respectively.

For an efficient calculation of HMM based word posterior probabilities, the forward-backward algorithm can be applied. This algorithm sums over all possible target sentences (of length I) without explicitly listing them. The formula given in equation 3.17 requires

processing along the source positions j as well as along the target positions i. In order to apply the forward-backward algorithm, the processing has to be restricted to one of the axes. There are two ways of achieving this: one is to assume a **monotone** alignment a_1^J:

$$a_{j-1} \leq a_j, \quad 1 \leq j \leq J \ .$$

Since the word alignment is not monotone for most language pairs, either the source or the target sentence has to be reordered to obtain this monotonicity. A detailed description of confidence estimation using monotone HMMs will be given in section 3.3.2.2. The second way of achieving an efficient calculation is to consider the so-called **inverted** HMM. In this model, the alignment is considered as a mapping b_1^I of target to source positions:

$$\begin{array}{rccl} b: & \{1,\ldots,I\} & \rightarrow & \{0,\ldots,J\} \\ & i & \mapsto & b_i = j \ . \end{array}$$

Note that this view of alignments is different from the one introduced in section 1.1 which is the one commonly used in the literature [Brown et al. 93]. The inverted model requires the mapping to be a function of the target words, i.e. each target word is aligned to exactly one source word (including the empty word). Word posterior probabilities based on the inverted HMM will be presented in section 3.3.2.3.

In the equations throughout this section, a bigram language model will be used in order to keep the notation manageable. In the experiments, either bi- or trigram language models have been applied. Details will be given together with the experimental results.

3.3.2.2 Monotone HMM

As stated above, the monotone HMM aligns every source word to either exactly one target word or the empty word. The following equations will be presented for a model *without* empty word. This is sufficient to illustrate the concept of word posterior probabilities based on the monotone HMM and their calculation using dynamic programming. The integration of the empty word into the model will shortly be described at the end of this section.

To reduce the computational effort, the distance of target positions a_j and a_{j-1} is restricted to be no greater than a fixed value K:

$$0 \ \leq \ a_j - a_{j-1} \ \leq \ K \tag{3.18}$$

This implies that no more than $K-1$ unaligned target words may follow each other. This holds also for the sentence boundary, i.e.

$$a_1 \leq K-1, \quad \text{and} \quad a_J \geq I-(K-1) \ .$$

Note that the maximum target sentence length is now

$$I_{max} = (J+1) \cdot K \ - \ 1 \ .$$

Choosing $K = 2$ yields the so-called 0-1-2-model which is illustrated in figure 3.2. When translating the source word in position j by the target word in position $a_j = i$, the

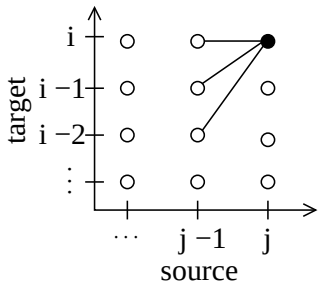


Figure 3.2: Example of possible transitions in a 0-1-2-HMM.

predecessor position a_{j-1} has to be in the set $\{i-2, i-1, i\}$. In the experiments, $K = 3$ has been used, but the equations will be given for $K = 2$.

The monotone 0-1-2-model provides a coverage constraint on the source words. This is different from the inverted HMM (see section 3.3.2.3) which does not guarantee to cover all source positions. Moreover, the restriction of the jump width in equation 3.18 limits the number of target words which occur unaligned. The monotone HMM is thus well-suited for language pairs with a similar number of words in both languages.

In order to handle the unaligned target words in the language model, an auxiliary quantity is introduced to account for all words between positions a_{j-1} and a_j. Let $k := a_j - a_{j-1}$. The language model probability of the $k-1$ inserted words can then be calculated recursively. For a bigram LM, this reads as follows:

$$p_k(e \,|\, e') := \begin{cases} p(e \,|\, e') & \text{if } k = 1 \\ \sum_{\tilde{e}} p(e \,|\, \tilde{e}) \cdot p_{k-1}(\tilde{e} \,|\, e') & \text{if } 1 < k \leq K \\ 0 & \text{if } k > K \end{cases} \tag{3.19}$$

Because the summation over all possible target sentences is performed, it is not specified which words $\tilde{e}$ may occur unaligned in equation 3.19. It is computationally infeasible to allow for every word in the vocabulary. Besides, it would not be sensible as most of these words have no connection with the source sentence. Thus for each target word e aligned to source word f_j, a set $V(e, f_j)$ of target words is defined which can be inserted before e. Using information from the word translation lexicon, this set is specified to contain all target words which

- have non-zero probability of being aligned to the empty word or
- are possible translations of source word f_j.

The auxiliary language model quantity is then defined as

$$p_k(e \,|\, e') := \begin{cases} p(e \,|\, e') & \text{if } k = 1 \\ \sum_{\tilde{e} \in V(e, f_j)} p(e \,|\, \tilde{e}) \cdot p_{k-1}(\tilde{e} \,|\, e') & \text{if } 1 < k \leq K \\ 0 & \text{if } k > K \end{cases}$$

Using the quantity $p_k(e \mid e')$, the joint sentence probability $p(e_1^I, f_1^J)$ in equation 3.17 can be rewritten as follows:

$$\begin{aligned} p(e_1^I, f_1^J) &= \prod_i p(e_i \mid e_{i-1}) \cdot \sum_{a_1^J} \prod_j p(a_j \mid a_{j-1}, I) \cdot p(f_j \mid e_{a_j}) \\ &= \sum_{a_1^J} \prod_i p(e_i \mid e_{i-1}) \cdot \prod_j p(a_j \mid a_{j-1}, I) \cdot p(f_j \mid e_{a_j}) \\ &= \sum_{a_1^J} \prod_j p(a_j \mid a_{j-1}, I) \cdot p(f_j \mid e_{a_j}) \cdot p_{a_j - a_{j-1}}(e_{a_j} \mid e_{a_{j-1}}) \ . \end{aligned} \tag{3.20}$$

At the beginning of the sentence, an artificial position 0 is introduced in order to obtain a proper definition of all probabilities. For this position, let $a_0 = 0$ and $e_{a_0} =< s >$, the sentence start symbol.

To determine word posterior probabilities based on a monotone HMM, the sentence probability given in equation 3.20 has to be normalized. As mentioned in section 3.1, it has to be divided by the sum of the probabilities of all competing target sentences $\tilde{e}_1^I$:

$$p(e_1^I \mid f_1^J, I) = \frac{p(e_1^I, f_1^J)}{\sum_{\tilde{e}_1^I} p(\tilde{e}_1^I, f_1^J)} \ .$$

Using the monotone HMM, word posterior probabilities can then be computed along the $j-$axis. The sum in equation 3.20 can be split into partial sums and dynamic programming can be applied. In the following, the dynamic programming equations for a 0-1-2-model with a bigram LM will be derived. Subsequently, the extension of these equations to include the empty word will be shortly discussed. In order to keep the presentation simple, the extension to a monotone HMM with $K > 2$ and a trigram language model will not be shown in this thesis, the interested reader is referred to [Schoenemann 05].

For the calculation of the word posterior probability $p_i(e \mid f_1^J)$ of word e occurring in position i, an auxiliary quantity is introduced: Let

$$B_i \subseteq \{1, \ldots, J\}$$

be the set of all source positions aligned to i. Then, the word posterior probability is given as the sum over all sentences e_1^I and over alignments a_1^J where only the positions contained in B_i are aligned to i, that is $a_j = i \iff j \in B_i$. As long as a model without empty word is considered, B_i is a compact set. The word posterior probability can then be determined as

$$\begin{aligned} p_i(e \mid f_1^J, I) &= \sum_{e_1^I} \delta(e_i, e) \cdot p(e_1^I \mid f_1^J, I) \\ &= \sum_{e_1^I} \delta(e_i, e) \sum_{B_i} \sum_{a_1^J} \delta(a_j = i \iff j \in B_i) \ \cdot \ p(e_1^I, a_1^J \mid f_1^J, I) \ , \end{aligned}$$

where $\delta(\cdot)$ denotes the extension of the Kronecker delta as introduced before (see appendix E). Note that the length I of the target sentence is kept fixed.

It is important to sum over all sets B_i instead of all source positions j aligned to i. The sets B_i form a disjoint decomposition of the cases over which the summation is performed. Every set B_i must be included exactly once in the word posterior probability. In the summation, two different cases are thus distinguished: B_i can be empty or non-empty. Since the actual processing is performed along source positions j rather than along the sets B_i, the sum is reformulated by considering the last source position j which is aligned to i. This leads to

$$\begin{aligned}
& p_i(e \,|\, f_1^J, I) = \qquad (3.21)\\
&= \sum_{e_1^I} \delta(e_i, e) \sum_{B_i} \sum_{a_1^J} \delta(a_j = i \iff j \in B_i) \cdot p(e_1^I, a_1^J \,|\, f_1^J, I) \\
&= \sum_{B_i \neq \emptyset} \sum_{a_1^J} \delta(a_j = i \iff j \in B_i) \sum_{e_1^I} \delta(e_i, e) \cdot p(e_1^I, a_1^J \,|\, f_1^J, I) \\
&\quad + \sum_{a_1^J} \delta(\forall j \;\; a_j \neq i) \sum_{e_1^I} \delta(e_i, e) \cdot p(e_1^I, a_1^J \,|\, f_1^J, I) \\
&= \sum_{j} \sum_{a_1^J} \delta(a_j = i \wedge \forall j' > j \,:\, a_{j'} > i) \; p(e_1^I, a_1^J \,|\, f_1^J, I) \\
&\quad + \sum_{a_1^J} \delta(\forall j \;\; a_j \neq i) \sum_{e_1^I} \delta(e_i, e) \cdot p(e_1^I, a_1^J \,|\, f_1^J, I) \; .
\end{aligned}$$

Now, the calculation can be easily performed along the source positions j. In the case that B_i is non-empty, only those alignments a_1^J are considered where j is the last source position aligned to i. Thus, the sum is split into disjoint parts. The alignment $a_j = i$ is kept fixed, but there is no restriction on the alignment a_1^{j-1} of the predecessor source words. Through this reformulation, the calculation can be divided into a forward and a backward part.

The integration of the probabilities given in equation 3.20 into equation 3.21 will lead to a representation from which forward and backward quantities can be derived. With the normalization term

$$p(f_1^J) = \sum_{\tilde{e}_1^I} p(\tilde{e}_1^I, f_1^J) \; ,$$

the word posterior probability can be written in the following way:

$$\begin{aligned}
p_i(e \,|\, f_1^J, I) = \frac{1}{p(f_1^J)} \cdot & \\
\cdot \Bigg[\sum_j \sum_{a_1^j} \delta(a_j, i) \sum_{e_1^i} \delta(e_i, e) \prod_{j'=1}^{j} & p(a_{j'} \,|\, a_{j'-1}, I) \cdot p(f_{j'} \,|\, e_{a_{j'}}) \cdot p_{a_{j'} - a_{j'-1}}(e_{a_{j'}} \,|\, e_{a_{j'-1}}) \\
\cdot \sum_{a_j^J} \delta(a_j = i \wedge \forall j' > j \,:\, a_{j'} > i) & \sum_{e_i^I} \delta(e_i, e) \cdot \\
\prod_{j'=j+1}^{J} & p(a_{j'} \,|\, a_{j'-1}, I) \cdot p(f_{j'} \,|\, e_{a_{j'}}) \cdot p_{a_{j'} - a_{j'-1}}(e_{a_{j'}} \,|\, e_{a_{j'-1}})
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_j p(i+1 \,|\, i-1, I) \\
&\cdot \sum_{a_1^j} \delta(a_j, i-1) \sum_{e'} \sum_{e_1^{i-1}} \delta(e_{i-1}, e') \cdot p(e \,|\, e') \cdot \\
&\qquad \cdot \prod_{j'=1}^{j} p(a_{j'} \,|\, a_{j'-1}, I) \cdot p(f_{j'} \,|\, e_{a_{j'}}) \cdot p_{a_{j'}-a_{j'-1}}(e_{a_{j'}} \,|\, e_{a_{j'-1}}) \\
&\cdot \sum_{a_{j+1}^J} \delta(a_{j+1}, i+1) \sum_{e''} \sum_{e_{i+1}^I} \delta(e_{i+1}, e'') \cdot p(e'' \,|\, e) \cdot p(f_{j+1} \,|\, e_{i+1}) \\
&\qquad \cdot \prod_{j'=j+2}^{J} p(a_{j'} \,|\, a_{j'-1}, I) \cdot p(f_{j'} \,|\, e_{a_{j'}}) \cdot p_{a_{j'}-a_{j'-1}}(e_{a_{j'}} \,|\, e_{a_{j'-1}}) \Bigg] \; .
\end{aligned}
$$

The first three lines represent the case that B_i is non-empty. In this sum, the backward part is restricted to alignments which do *not* include target position i. The remaining five lines of the equation above correspond to the case $B_i = \emptyset$. The source word in position j has to be aligned to $i-1$, and that in position $j+1$ to i, respectively. So in the forward part, it is required that $a_j = i-1$. For the backward part, $a_{j+1} = i+1$ must hold. Apart from that, the backward calculation is not restricted and includes all possible alignments a_{j+2}^J.

From this representation, forward and backward quantities can be deducted. The forward quantity $Q^F(i,j,e)$ reflects the probability of translating f_1^j as e_1^i given that $a_j = i$. The backward quantity $Q^B(i,j,e)$ expresses the probability of translating f_{j+1}^J as e_{i+1}^I given that $a_j = i$.

If B_i is non-empty, an additional quantity $Q^{BR}(i,j,e)$ is needed, called *restricted backward* probability here. It additionally enforces that $a_{j'} > i$ for $j' > j$. These quantities are defined by

$$
\begin{aligned}
Q^F(i,j,e) &:= \sum_{e_1^i} \delta(e_i, e) \sum_{a_1^j} p(e_1^i, a_1^j \,|\, f_1^j, a_j = i) \\
Q^B(i,j,e) &:= p(f_j \,|\, e) \cdot \sum_{e_{i+1}^I} \sum_{a_{j+1}^J} p(e_{i+1}^I, a_{j+1}^J \,|\, f_{j+1}^J, a_j = i) \\
Q^{BR}(i,j,e) &:= p(f_j \,|\, e) \cdot \sum_{e_{i+1}^I} \sum_{a_{j+1}^J} \delta(\forall j' > j \,:\, a_{j'} > i) \cdot p(e_{i+1}^I, a_{j+1}^J \,|\, f_{j+1}^J, a_j = i) \; .
\end{aligned}
$$

These forward and backward quantities can be computed recursively over j and i. Since the underlying model is a 0-1-2-model, the predecessor position of i has to be in the set

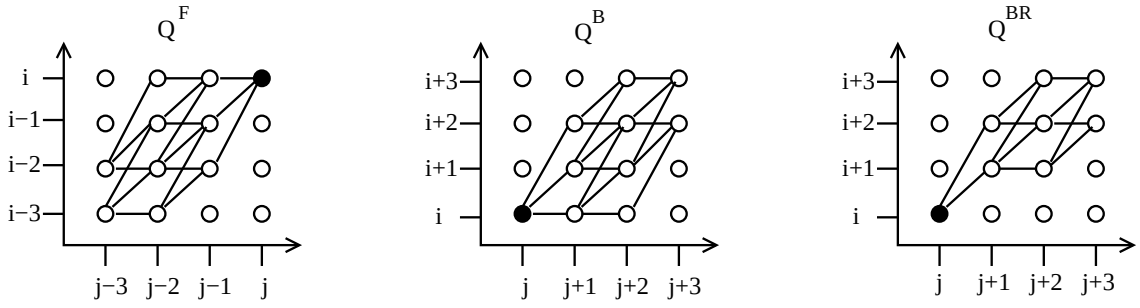


Figure 3.3: The three grid structures needed for confidence estimation with a monotone HMM: the forward grid, the backward grid, and the restricted backward grid. The shown paths are included in the score for the filled grid point in a 0-1-2-model, where $J = 4$ and $I = 4$.

$\{i-2, i-1, i\}$. The recursion is the following:

$$
\begin{aligned}
Q^F(i,j,e) &= \sum_{a_1^j} \delta(a_j, i) \sum_{e_1^i} \delta(e_i, e) \prod_{j'=1}^{j} p(a_{j'} \,|\, a_{j'-1}, I) \cdot p(f_{j'} \,|\, e_{a_{j'}}) \cdot p_{a_{j'}-a_{j'-1}}(e_{a_{j'}} \,|\, e_{a_{j'-1}}) \\
&= p(f_j \,|\, e) \cdot \sum_{k=0}^{2} p(i \,|\, i-k, I) \cdot \sum_{e'} p_k(e \,|\, e') \cdot Q^F(i-k, j-1, e') \\
Q^B(i,j,e) &= p(f_j \,|\, e) \cdot \sum_{a_j^J} \delta(a_j, i) \sum_{e_i^I} \delta(e_i, e) \prod_{j'=j+1}^{J} p(a_{j'} \,|\, a_{j'-1}, I) \cdot p(f_{j'} \,|\, e_{a_{j'}}) \cdot \\
&\qquad \cdot p_{a_{j'}-a_{j'-1}}(e_{a_{j'}} \,|\, e_{a_{j'-1}}) \\
&= p(f_j \,|\, e) \cdot \sum_{k=0}^{2} p(i+k \,|\, i, I) \cdot \sum_{e'} p_k(e' \,|\, e) \cdot Q^B(i+k, j+1, e') \ .
\end{aligned}
$$

The calculation of the restricted backward quantity is different, because this is not recursive in itself, but depends on the general backward quantity. The jump with k has to be greater than zero in this case:

$$
\begin{aligned}
Q^{BR}(i,j,e) &= p(f_j \,|\, e) \cdot \sum_{a_j^J} \delta(a_j = i \wedge \forall j' > j \ : \ a_{j'} > i) \sum_{e_i^I} \delta(e_i, e) \cdot \\
&\qquad \cdot \prod_{j'=j+1}^{J} p(a_{j'} \,|\, a_{j'-1}, I) \cdot p(f_{j'} \,|\, e_{a_{j'}}) \cdot \ p_{a_{j'}-a_{j'-1}}(e_{a_{j'}} \,|\, e_{a_{j'-1}}) \\
&= p(f_j \,|\, e) \cdot \sum_{k=\mathbf{1}}^{2} p(i+k \,|\, i, I) \cdot \sum_{e'} p_k(e' \,|\, e) \cdot Q^{\mathbf{B}}(i+k, j+1, e') \ .
\end{aligned}
$$

Figure 3.3 depicts how these quantities can be visualized in a grid. The selected grid point in the plots is (j, i), represented by the filled point (located in the upper right corner in the left plot, and in the lower left corner in the other plots, respectively). The

principle holds for all grid points and sentence lengths. There is an invisible third axis for the target words e. The left plot shows all incoming paths which are considered for the forward probability. It reflects the restrictions of the 0-1-2-model: it is not possible to jump from target position $i-3$ directly to i. The difference between the backward quantities Q^B and Q^{BR} is depicted in the other two plots: All paths where target position i is visited again are excluded from the restricted backward quantity.

These forward and backward probabilities are combined into a word posterior probability of word e occurring in position i. Again, the two cases that e is aligned to a source word f_j and that it occurs unaligned have to be handled separately. This yields:

$$p_i(e \,|\, f_1^J, I) = \frac{1}{p(f_1^J)} \left[\sum_j \frac{Q^F(i,j,e) \cdot Q^{BR}(i,j,e)}{p(f_j \,|\, e)} + p(i+1 \,|\, i-1, I) \sum_{e'} p(e \,|\, e') \sum_j Q^F(i-1,j,e') \sum_{e''} p(e'' \,|\, e) \cdot Q^B(i+1,j+1,e'') \right] .$$

The normalization constant $p(f_1^J)$ can be computed as follows:

$$p(f_1^J) \;=\; \sum_{\tilde{e}_1^I} p(\tilde{e}_1^I, f_1^J) = \sum_e Q^F(I,J,e) \; .$$

Analogously to the computation over word graphs, the sum of the probabilities of all possible target sentences is equivalent to the sum over all forward probabilities (see section 3.2.1.1).

Monotone HMM with empty word

The monotone HMM can be extended by the introduction of an empty target word which the source words may align to. The concept presented at the beginning of section 3.3.2 modifies the alignment probabilities by introducing artificial target positions for empty words. The possible transitions from target position $a_j = i$ to $a_{j+1} = i'$ in the 0-1-2-model will then be:

$$\begin{array}{lll} i \;\rightarrow\; i' \in \{i, i+1, i+2, i+I\} & & 1 \leq i \leq I \\ i \;\rightarrow\; i' \in \{i, (i-I)+1, (i-I)+2\} & & I+1 \leq i \leq 2 \cdot I \end{array}$$

After visiting an empty word in position $i+I$, the next target word has to be either the same empty word, or one of the actual target words following position i. These additional cases have to be included in the equations. The modified equations will not be presented in this thesis, for details see [Schoenemann 05].

Reordering of source and target sentence

The monotone HMM requires the word alignment between source and target sentence to be monotone. Since this is normally not the case for natural languages, either source or target sentence has to be reordered to match the order of the other one.

If the *source* sentence is reordered, the most likely word alignment between the sentences using the IBM model 4 alignment model is determined. Afterwards, the source sentence is reordered such that this alignment becomes monotone. Unaligned source words are placed right after the preceding aligned source word. The reordered source sentence is taken as input for the confidence estimation.

If instead the *target* sentences are reordered, a new language model has to be trained on these data. Thus, the same reordering is applied to the training corpus, and a new language model is trained. On development and test set, N-best lists of alignments are generated. For each of these alignments the resulting reordered target sentence is computed. This allows to estimate reordering probabilities $p_r(i' \mid i, e_1^I, f_1^J)$ that the word in position i is reordered to position i'. These probabilities are estimated for a specific sentence pair simply by counting how often the word occurs in the respective position. Now the confidence of target word e occurring in position i in the reordered sentence can be computed as

$$p_i^r(e \mid f_1^J) \;=\; \sum_{i'} p_r(i' \mid i, e_1^I, f_1^J) \cdot p_{i'}(e \mid f_1^J) \;.$$

Since the reordering of the target sentence yields better results for confidence estimation, only these experiments will be reported here.

3.3.2.3 Inverted HMM

Another way of handling the calculation of word posterior probabilities using an HMM is to consider an **inverted** alignment model b_1^I as introduced on page 27:

$$\begin{array}{rccl} b: & \{1,\ldots,I\} & \rightarrow & \{0,\ldots,J\} \\ & i & \mapsto & b_i = j \;. \end{array}$$

This maps target positions to source positions with the constraint that each target word cannot be aligned to more than one source word. Similar to the monotone model, dynamic programming can be applied to determine word posterior probabilities using the inverted model. Again, the forward-backward algorithm is used to efficiently calculate the relevant quantities.

The main disadvantage of the inverted HMM is that it does not provide a coverage constraint. There is no way to enforce that all source positions are covered by the alignment b_1^I. The confidence measures based on the inverted HMM suffer from this. They do not perform well as the experimental results presented in section 7.2.2 will show. Therefore, the approach will be shortly described in this section, but the equations are not given in detail. The interested reader is referred to [Schoenemann 05].

The sentence probability for the inverted HMM in combination with a bigram language

model is given by

$$\begin{aligned} p(e_1^I, f_1^J) &= p(e_1^I) \cdot p(f_1^J | e_1^I) \\ &= p(e_1^I) \cdot \sum_{b_1^I} p(f_1^J, b_1^I | e_1^I) \\ &= \prod_i p(e_i | e_{i-1}) \cdot \sum_{b_1^I} \prod_i p(b_i | b_{i-1}, J) \cdot p(f_{b_i} \,|\, e_i) \\ &= \sum_{b_1^I} \prod_i p(e_i | e_{i-1}) \cdot p(b_i | b_{i-1}, J) \cdot p(f_{b_i} \,|\, e_i) \ . \end{aligned}$$

Note that only the alignment, but not the lexicon model is inverted. The advantage of introducing the inverted alignment model is that now the word posterior probabilities can efficiently be calculated along the target positions i.

As described before, the sentence probability is normalized to obtain the posterior:

$$p(e_1^I \,|\, f_1^J) = \frac{1}{p(f_1^J)} \cdot p(e_1^I, f_1^J) \ , \quad \text{with} \quad p(f_1^J) = \sum_{\tilde{e}_1^I} p(\tilde{e}_1^I, f_1^J) \ .$$

The word posterior probability of target word e occurring in sentence position i is then determined as

$$\begin{aligned} p_i(e \,|\, f_1^J) &= \sum_{e_1^I} \delta(e_i, e) \cdot p(e_1^I \,|\, f_1^J) \\ &= \frac{1}{p(f_1^J)} \cdot \sum_{e_1^I} \delta(e_i, e) \sum_{b_1^I} \prod_{i'} p(e_{i'} | e_{i'-1}) \cdot p(b_{i'} | b_{i'-1}, J) \cdot p(f_{b_{i'}} \,|\, e_{i'}) \ . \end{aligned}$$

As this equation shows, forward-backward calculation can be applied to the product over the target positions i. The basic idea behind the computation is to keep the alignment $b_i = j$ fixed. The probability of e given this alignment point can then be split into a forward and a backward quantity. These two parts can be calculated recursively. The product of forward and backward score, with a proper normalization, yields $p_i(e, j | f_1^J, I)$. The word posterior probability of e occurring in target position i is obtained by summing these quantities over all possible alignments $j = b_i$:

$$p_i(e | f_1^J, I) = \sum_j p_i(e, j | f_1^J, I) \ .$$

3.3.3 Direct confidence measures based on phrases

In this section, the use of phrase-based translation models for the calculation of word posterior probabilities will be explained. First, the underlying phrase-based translation approach will be depicted in section 3.3.3.1. In section 3.3.3.2, its use for confidence estimation will be explained and word-level confidence measures will be derived. Section 3.3.3.3 will systematically analyze the differences between the *direct* phrase-based approach and the *system-based* confidence measures applied to output from the phrase-based translation system.

3.3.3.1 Phrase-based translation approach (PBT)

This section contains a brief description of the phrase-based statistical machine translation system. A detailed description can be found in [Zens and Ney 04, Zens and Ney 05]. The key elements of this translation approach are bilingual phrases, i.e. pairs of source and target language phrases, where a phrase is simply a contiguous sequence of words. These bilingual phrases are extracted from a word-aligned bilingual training corpus. GIZA++ is used to train this word alignment[a]. To obtain a more symmetric word alignment, the training is performed in both translation directions and the resulting Viterbi alignments are unified [Och and Ney 03].

In this translation approach, the posterior probability $Pr(e_1^I \mid f_1^J)$ is modeled directly using a weighted log-linear combination of a trigram language model and various translation models: a phrase translation model and a word-based lexicon model. These translation models are used for both directions: $p(f \mid e)$ and $p(e \mid f)$. Additionally, a word penalty and a phrase penalty are used. With the exception of the language model, all models can be considered as within-phrase models as they depend only on a single phrase pair, but not on the context outside the phrase.

The monotone search algorithm from [Zens and Ney 04] has been extended such that reorderings are possible. The reordering is performed on two levels: on the word and on the phrase level. For the word-level reordering, all possible permutations of the source positions (under certain restrictions) are allowed. These permutations are represented as a reordering graph, similar to [Zens et al. 02]. Once this reordering graph is given, a phrase-based translation of this graph can be performed. More details of this reordering approach are described in [Kanthak et al. 05]. Additionally, reordering on the phrase level can take place, i.e. whole blocks of words can be reordered.

Here, the equations for the monotone search will be given in order to keep the presentation simple. The extension to the non-monotone case is straightforward. Let (j_0^K, i_0^K) be a monotone segmentation of the sentence pair into phrases, with the corresponding (bilingual) phrase pairs $(\tilde{f}_k, \tilde{e}_k) = (f_{j_{k-1}+1}^{j_k}, e_{i_{k-1}+1}^{i_k})$, $k = 1, \ldots, K$. The phrase-based approach to SMT is then expressed by the following equation:

$$\begin{aligned} \hat{e}_1^{\hat{I}} \;=\; & \underset{K, j_0^K, i_0^K, I, e_1^I}{\operatorname{argmax}} \left\{ \prod_{i=1}^{I} \left[e^{\lambda_1} \cdot p(e_i \mid e_{i-2}^{i-1})^{\lambda_2} \right] \right. \\ & \left. \cdot \prod_{k=1}^{K} \left[\; e^{\lambda_3} \cdot \; p(\tilde{f}_k \mid \tilde{e}_k)^{\lambda_4} \cdot p(\tilde{e}_k \mid \tilde{f}_k)^{\lambda_5} \cdot \prod_{j=j_{k-1}+1}^{j_k} p(f_j \mid \tilde{e}_k)^{\lambda_6} \cdot \prod_{i=i_{k-1}+1}^{i_k} p(e_i \mid \tilde{f}_k)^{\lambda_7} \right] \right\}, \end{aligned} \tag{3.22}$$

where $p(\tilde{f}_k \mid \tilde{e}_k)$ and $p(\tilde{e}_k \mid \tilde{f}_k)$ are the phrase lexicon models in both translation directions. The phrase translation probabilities are computed as a log-linear interpolation of the relative frequencies and the IBM model 1 probability. The single word based lexicon models are denoted as $p(f_j \mid \tilde{e}_k)$ and $p(e_i \mid \tilde{f}_k)$, respectively. $p(f_j \mid \tilde{e}_k)$ is defined as the IBM model 1 probability of f_j over the whole phrase $\tilde{e}_k$, computed from the single-word-based probabilities, and $p(e_i \mid \tilde{f}_k)$ is the inverse model, respectively. e^{λ_1} is the so-

[a] The GIZA++ toolkit for word alignment can be downloaded from
`http://www-i6.informatik.rwth-aachen.de/web/Software/index.html`

called word penalty, and e^{λ_3} is the phrase penalty, assigning constant costs to each target language word/phrase. The language model is a trigram model with modified Kneser-Ney discounting and interpolation [Stolcke 02]. The search determines the target sentence and segmentation which maximize the objective function.

As equation 3.22 shows, the sub-models are combined via weighted log-linear interpolation. The model scaling factors $\lambda_1, \ldots, \lambda_7$ are optimized with respect to some MT evaluation criterion [Och 03] such as BLEU score.

3.3.3.2 Direct approach to confidence estimation using phrases

The statistical models presented in section 3.3.3.1 can be used to estimate the confidence of target words. In contrast to the approaches presented in section 3.2, the context information is not integrated at the sentence level, but only at the phrase level. A sort of marginal probability $Q(e, f_1^J)$ is determined. For estimating this probability, the phrase lexicon which has been built in training is used as knowledge source. All source phrases f_j^{j+s} which occur in the given source sentence f_1^J are extracted, for all possible phrase lengths $s+1$. For each such source phrase, the bilingual phrase lexicon contains the possible translations e_i^{i+t}. The confidence of target word e is then calculated by summing over all phrase pairs (f_j^{j+s}, e_i^{i+t}) where the target part e_i^{i+t} contains the word e.

Let $Q_{LM}(e_i^{i+t})$ be the language model score of the target phrase together with the word penalty e^{λ_1} for each word in the phrase , i.e.

$$Q_{LM}(e_i^{i+t}) \;:=\; \prod_{i'=i}^{i+t} e^{\lambda_1} \cdot p(e_{i'} \mid e_{i'-2}^{i'-1})^{\lambda_2} \;. \tag{3.23}$$

The language model probability at the phrase boundary is approximated by a unigram and bigram. Define $Q_{PM}(f_j^{j+s}, e_i^{i+t})$ as the score of the phrase pair which consists of the phrase penalty e^{λ_3}, the phrase lexicon scores, and the two word lexicon model scores (see section 3.3.3.1):

$$\begin{aligned} Q_{PM}(f_j^{j+s}, e_i^{i+t}) \;:=\; & e^{\lambda_3} \cdot\; p(f_j^{j+s} \mid e_i^{i+t})^{\lambda_4} \cdot p(e_i^{i+t} \mid f_j^{j+s})^{\lambda_5} \\ & \cdot \prod_{j'=j}^{j+s} p(f_{j'} \mid e_i^{i+t})^{\lambda_6} \cdot \prod_{i'=i}^{i+t} p(e_{i'} \mid f_j^{j+s})^{\lambda_7} \;. \end{aligned} \tag{3.24}$$

The (unnormalized) confidence of target word e is then determined by combining language model and phrase model score of all phrase pairs containing e:

$$Q(e, f_1^J) \;:=\; \sum_{j=1}^{J} \sum_{s=0}^{\min\{s_{\max}, J-j\}} \sum_{e_i^{i+t}} \delta(e \in e_i^{i+t}) \cdot Q_{LM}(e_i^{i+t}) \cdot Q_{PM}(f_j^{j+s}, e_i^{i+t}) \;,$$

where $s \leq s_{\max}$ and t are source and target phrase lengths, $s_{\max}$ being the maximal source phrase length.

The value calculated in equation 3.25 is not normalized. In order to obtain a probability, this value is divided by the sum over the (unnormalized) confidence values of all target

words:

$$p_{phr}(e \mid f_1^J) = \frac{Q(e, f_1^J)}{\sum_{e'} Q(e', f_1^J)} \ . \tag{3.25}$$

If the normalization is performed as shown in equation 3.25, a probability distribution over all target words given the source sentence is obtained. Thus, all words occurring in a target sentence compete with each other, no matter how long the sentence is. Moreover, several translation alternatives, which might all be correct, have to share the probability mass. In order to overcome the latter problem, a different normalization strategy has been investigated. The confidence is normalized by the probability mass of all possible target *phrases* rather than words. Then, the competing events are *"e is contained in the target sentence"* versus *"e is* not *contained in the target sentence"*. Define

$$Q(\neg\, e, f_1^J) \ := \ \sum_{j=1}^{J} \sum_{s=0}^{\min\{s_{\max}, J-j\}} \sum_{e_i^{i+t}} \left(1 - \delta(e \in e_i^{i+t})\right) \cdot Q_{LM}(e_i^{i+t}) \cdot Q_{PM}(f_j^{j+s}, e_i^{i+t}) \ .$$

The normalization is then performed as follows:

$$p_{phr}(c = 1 \mid e, f_1^J) = \frac{Q(e, f_1^J)}{Q(e, f_1^J) + Q(\neg\, e, f_1^J)} \ ,$$

which can be interpreted as the probability of e to be contained in a translation of f_1^J. Analogously, $p_{phr}(c = 0 \mid e, f_1^J)$ is defined as the probability of e *not* to occur.

Nevertheless, this type of normalization does not solve the problem that the longer the source sentence, the more target words should be classified as correct. A solution to this will be presented in section 4.5. The idea behind it is not to modify the probability assigned to the target word, but to modify the classifier by adapting the acceptance threshold.

In equation 3.25, the language model only determines the probability of the words within the target part of the phrase, and not across the phrase boundaries. Only the single target phrase e_i^{i+t} without context is considered. The words at the beginning of the phrase will only be scored by a unigram- or bigram-model. Since the phrases are extracted from the training corpus, they represent valid word sequences. Therefore, one can assume that the language model will not have much influence on the confidence estimation. Experiments using a direct phrase-based model without language model will be presented in section 7.2. Similar considerations hold for word and phrase penalty: In the translation process, they are useful for adjusting the length of the generated target hypothesis and for assigning more weight to longer phrases. Since this does not make much sense in the confidence estimation setting, confidence measures without word and phrase penalty have also been investigated.

Scaling factors

As shown in the equations in section 3.3.3.2, the different sub-models of the phrase-based translation approach are combined in a log-linear manner. The weights $\lambda_1, \ldots, \lambda_7$

are optimized in the translation process w.r.t. some evaluation criterion such as WER or BLEU. This is done using the Downhill Simplex algorithm [Press et al. 02]. The resulting values of the weights express the relation between the sub-models, but not their absolute values. They are usually normalized so that they sum up to 1. For the use in confidence estimation, two different aspects have thus to be considered:

- The relation of the sub-models which is optimal for translation quality is not necessarily optimal for classification performance. If the translation alternatives contained in an N-best list differ only in a small part of the sentence, their probabilities often lie close together. For the use in confidence estimation, a distribution which is less smooth is desirable to distinguish between good and bad translations. Therefore, the sub-model scaling factors are optimized w.r.t. some confidence error measure (see appendix 4.3). The direct phrase-based confidence measures provide a framework for optimizing the sub-model weights efficiently, which is not the case for the system-based confidence measures. The optimization is performed analogously to the procedure for machine translation: The confidence values are determined for all words in the development corpus. Then, classification is carried out as described in section 4.1, and the result is evaluated. The weights are then modified and the confidence estimation is repeated, until optimal classification performance on the development set is achieved. Again, the Downhill Simplex algorithm is used for optimization.

- Whereas for MT only the relation between the different sub-models, but not the actual values of the scaling factors are important, the word posterior probabilities used as confidence measures also depend on these actual values. In MT, the sub-model scaling factors are normalized such that they sum up to 1. For the use in confidence estimation, this sum can take on different values. That is, in addition to the individual value of each λ_i, the normalization value

$$\Lambda := \sum_{i=1}^{7} \lambda_i$$

  is optimized. This is done by choosing the value Λ and optimizing the λ_i accordingly. The procedure is carried out for several possible values of Λ so that the optimum for both Λ and the $\lambda_i, i = 1, \ldots, 7$ is found. This Λ is analogous to the global scaling factor introduced on page 17 for the system-based confidence measures.

3.3.3.3 Direct phrase-based versus system-based confidence measures

The direct phrase-based confidence measures presented in the previous subsection are derived from the phrase-based translation model. They use the same sub-models, and carry out similar summations. So, if the N-best list based methods are applied to output from the phrase-based translation system, the resulting confidence measures should be similar to the direct phrase-based ones. This raises the question which modifications are necessary to transform one approach into the other.

Consider the N-best list based methods as starting point. To be as close to the direct model as possible, the method which disregards the target position completely (see section 3.2.1.5) will be studied. The word posterior probability is determined by calculating a score $Q_n(e, f_1^J)$, $n = 0, 1, 2$ and normalizing it:

$$p(e \mid f_1^J) = \frac{Q_n(e, f_1^J)}{\sum_{e'} Q_n(e', f_1^J)} .$$

In the following, different ways of calculating $Q_n(e, f_1^J)$ will be presented. The starting point is the N-best list based approach which will be denoted as Q_0. The models Q_1 and Q_2 will be modified to bridge the gap between this system-based and the direct phrase-based approach. For the original N-best list based approach, the score is defined as

$$Q_0(e, f_1^J) := \sum_{i=1}^{I_{max}} \sum_{I \geq i} \sum_{e_1^I} \delta(e_i, e) \cdot Q(f_1^J, e_1^I) ,$$

where I_{max} is the maximal target sentence length. In the following, the sentence score calculated by the phrase-based translation system will be integrated into this equation. To keep the presentation understandable, this will be done for a monotone translation without reordering of words or phrases. The extension to non-monotone translation is possible. However, it will not yield any additional insight and is therefore omitted here.

Consider a segmentation (j_0^K, i_0^K) of source and target sentence with the corresponding bilingual phrase pairs $(f_{j_{k-1}+1}^{j_k}, e_{i_{k-1}+1}^{i_k})$, $k = 1, \ldots, K$. The language and phrase model contribution are defined as given in equations 3.23 and 3.24. The score $Q_0(e, f_1^J)$ is then calculated as

$$Q_0(e, f_1^J) = \sum_{i=1}^{I_{max}} \sum_{I \geq i} \sum_{e_1^I} \delta(e_i, e) \cdot Q_{LM}(e_1^I) \sum_{K, j_0^K, i_0^K} \prod_{k=1}^{K} Q_{PM}(f_{j_{k-1}+1}^{j_k}, e_{i_{k-1}+1}^{i_k}) .$$

The first step in bringing the models closer together is to omit the context outside the phrase. For each target position i and each target sentence e_1^I, only the phrase pair covering e_i is considered. The language model at the phrase boundary is approximated via a bigram and a unigram model. This yields

$$\begin{aligned} Q_1(e, f_1^J) := \sum_{i=1}^{I_{max}} \sum_{I \geq i} \sum_{e_1^I} \delta(e_i, e) \sum_{K, j_0^K, i_0^K} \sum_{k=1}^{K} \delta(i_{k-1} + 1 \leq i \leq i_k) \cdot \\ \cdot Q_{LM}(e_{i_{k-1}+1}^{i_k}) \cdot Q_{PM}(f_{j_{k-1}+1}^{j_k}, e_{i_{k-1}+1}^{i_k}) . \end{aligned}$$

In this summation, the same phrase pair can be taken into account several times, if it occurs in more than one N-best list hypothesis. If this is changed, and each phrase pair is considered only once, the summation is equivalent to that carried out in the direct model:

$$\begin{aligned} Q_2(e, f_1^J) &:= \sum_{K, j_0^K} \sum_{k=1}^{K} \sum_{\tilde{e}} \delta(e \in \tilde{e}) \cdot Q_{LM}(\tilde{e}) \cdot Q_{PM}(f_{j_{k-1}+1}^{j_k}, \tilde{e}) \\ &= \sum_{j=1}^{J} \sum_{s=0}^{\min\{s_{\max}, J-j\}} \sum_{\tilde{e}} \delta(e \in \tilde{e}) \cdot Q_{LM}(\tilde{e}) \cdot Q_{PM}(f_j^{j+s}, \tilde{e}) , \end{aligned}$$

where s_{max} is the maximal source phrase length. Note that here the hypothesized target sentence e_1^I is not regarded anymore.

Following the considerations presented above, comparative experiments have been carried out to analyze the differences between the two approaches. Their results will be reported in section 7.2.6.

4 Word posterior probabilities as confidence measures

In this chapter, it will be explained how the word posterior probabilities introduced in this thesis can be used for confidence estimation. Section 4.1 will describe how to build a classifier which accepts or discards single words in a translation hypothesis based on their confidence. In order to evaluate such a classifier, the true classes of the words have to be given. Since there is no unique way of defining these true classes for MT output, section 4.2 will present several different possibilities. The evaluation measures which are used to assess the classification performance of the confidence measures will be introduced in section 4.3. Section 4.4 will describe methods which combine different confidence features, including a short overview of the experiments which were performed at the CLSP summer workshop 2003. Alternative ways of defining a classifier using confidence measures will be discussed in section 4.5.

4.1 Classification

The idea behind word-level confidence estimation is to be able to detect possible errors in the output of a machine translation system. Using confidence measures, individual words can be labeled as either *correct* or *incorrect*. This additional information can be used e.g. in interactive trans-type style machine translation systems as described in section 6.

Two problems have to be solved in order to compute confidence measures. First, suitable confidence features have to be computed. Second, a binary classifier has to be defined which decides whether a word is correct or not. The different types of word posterior probabilities introduced in chapter 3 can be interpreted as the probability of a word being correct. That is, the probability can directly be used as confidence measure. To this purpose, it is compared to a threshold t. All words whose confidence is above this threshold are tagged as correct and all others are tagged as incorrect translations. Thus, the binary classifier is defined as

$$\text{class}(e) = \begin{cases} \text{correct} & \text{if } p(e \mid f_1^J) \geq t \\ \text{incorrect} & \text{otherwise} \end{cases} \tag{4.1}$$

The threshold t is optimized on a distinct development set beforehand.

4.2 Word error measures

In order to evaluate the classifier described above, reference tags are needed which define the true class of each word. In machine translation, it is not intuitively clear how to determine the true class of a word. Therefore, a number of different measures for identifying the reference classes for single words in a translation hypothesis have been implemented. They are inspired by different translation evaluation measures like WER, PER, and BLEU. All of them compare the translation to one or – if available – several references to determine the word errors. The investigated measures are the following:

- **Pos**: This error measure considers a word as correct if it occurs in exactly the same target position as in one of the reference translations.
- **WER**: A word is counted as correct if it is Levenshtein-aligned to itself in one of the references.
- **PER**: A word is tagged as correct if it occurs in one of the reference translations. Here, the reference is regarded as a bag of words, i.e. the number of occurrences per word is taken into account. The position of the word in the sentence is completely disregarded.
- **Set**: This is a less strict variant of PER: The number of occurrences per word is not considered, i.e. a word occurring in the translation three times is tagged as correct every time, even if the reference contains it only once or twice.
- **n-gram**: This measure considers the word as well as its $n-1$ predecessors in the hypothesis. Only those words are labeled as correct which occur in the references together with this history. n has been chosen to be 2, 3, and 4. The references are pooled, analogous to the computation of BLEU or NIST score.

All word error measures except for "n-gram" exist in two variants: First, each translation hypothesis is compared to the pool of all references (in case that there exist different reference translations for the development and test corpus). Second, the reference with minimum distance to the hypothesis according to the translation evaluation measure under consideration is determined. The true classes of the words are then defined with respect to this reference. For example, if the metric PER is applied, the pooled variant labels all those words as correct that occur in any of the references. The second variant considers only those words as correct that are contained in the nearest reference. The latter corresponds to the procedure used for m-WER and m-PER in MT evaluation [Nießen et al. 00].

The symbol sequences in figure 4.1 illustrate the differences between the word error measures introduced above: Comparing the strings "ABCBDB" and "ABBCE", only two words are correct w.r.t. the word error measure Pos. WER additionally labels the 'C' as correct. According to PER, also the second 'B' is aligned to a 'B' in the reference. The word error metric Set also labels the third 'B' as correct, because the number of occurrences in the reference is completely disregarded. Thus, the number of hypothesis words which are labeled as correct varies between 2 and 5 for these error measures.

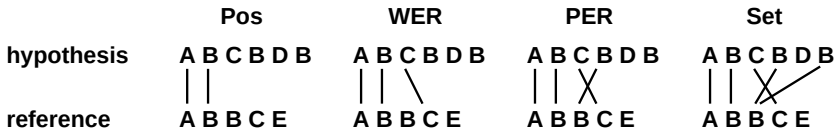


Figure 4.1: Example of the different word error measures: The string "ABCBDB" is compared to "ABBCE". Correct symbols are aligned.

Table 4.1 shows the percentage of words that are labeled as correct according to the different error measures on the training, development and test corpora of the EPPS task described in appendix A.2. It can be seen that Pos is a very pessimistic measure and considers only between a third and a half of the words correct. The other measures count much more words as correct because they do not require them to occur in the exact position of the reference. The introduction of the Levenshtein alignment into the error measure yields a large increase in the number of words considered correct. This number grows further the more the error criterion is relaxed (see PER, Set). Naturally, the n-gram metric becomes stricter the longer the history is: The ratio of correct words drops by about 20% absolute for the 4-gram versus the 2-gram. For the error measures existing in two variants, both the pool of the references (p) and the nearest reference (n) are considered in the table. The pooling yields a significant increase in the number of words labeled as correct. Note that those figures are not the translation errors for the system output. They are calculated only for the words contained in the generated translation hypotheses, and they are also normalized by the hypothesis lengths. The actual WER, for example, counts deletions as well and is normalized by the number of reference words.

Table 4.1: Ratio of correct words [%] in the EPPS Spanish → English development and test corpora according to different word error measures. Both the ratio based on the pooled (p) and on the nearest (n) reference are given.

Error Measure	Pos		WER		PER		Set		n-gram		
	p	n	p	n	p	n	p	n	2	3	4
DEV	49.3	47.0	78.6	72.9	81.5	77.4	83.0	79.2	61.2	50.0	42.8
TEST	33.7	31.1	76.5	69.8	81.5	76.5	84.6	80.3	55.7	42.3	33.4

4.3 Evaluation of confidence measures

For confidence estimation, one is interested in measuring how well a model discriminates between correct and incorrect translations. There exist two different ways of measuring this: One is to set a threshold for the acceptance of translations based on their confidence as described in section 4.1. This threshold is optimized on a development set and classification performance is evaluated on a test set. A metric suited for this setting is the classification error rate or confidence error rate (CER).

Evaluating the confidence measure in the abstract, independent of a fixed threshold, requires techniques that capture classification performance across the range of all possible thresholds. In this section, two such techniques will be described: ROC curves, and IROC. A measure which assesses not only the classification performance of the word posterior probabilities, but evaluates the actual probability estimates, is the so-called normalized cross entropy which will shortly be discussed in this section. It will be explained why this measures has not been used to evaluate the confidence measures presented in this thesis.

4.3.1 Classification error rate (CER)

The most intuitive measure is simply the proportion of errors made by the classifier over the test corpus. Consider a threshold t, and a decision procedure that accepts translations for which the confidence $c(e)$ is above the threshold t (see equation 4.1). The classification error rate is defined as the number of incorrect decisions divided by the total number of generated words in the translated sentence:

$$\begin{aligned} \text{CER} &= \frac{\#\text{incorrectly assigned tags}}{\#\text{generated words}} \\ &= \frac{\#\text{false acceptances} + \#\text{false rejections}}{\#\text{generated words}} \,. \end{aligned}$$

The baseline CER is determined by assigning the most frequent class (determined over all words in the considered corpus) to all translations. If the correct classes of the words are defined on the basis of WER and if the most frequent class is "correct", this is the number of substitutions and insertions, divided by the number of generated words:

$$\text{baseline CER} = \frac{\#\text{substitutions and insertions}}{\#\text{generated words}} \,.$$

The quality of a confidence measure can now be assessed by comparing the classification error rate with the baseline CER and measuring the improvement.

The CER as an evaluation criterion has several drawbacks. First, the two different types of errors are not distinguished. As a result, the CER depends on the prior probability of the two classes "correct" and "incorrect". If one is interested in one of these types of errors in particular, the CER is obviously not suitable. Second, the CER strongly depends on the choice of the tagging threshold. In order to get a realistic impression of the classifier's performance, the tagging threshold is adjusted beforehand on a development corpus distinct from the test set. The CER is used in this thesis for evaluation of confidence measures because of its simplicity, but since it measures the system performance for only one operating point, it should not be used as the only evaluation criterion.

4.3.2 ROC curves and IROC

The ROC curve plots the *correct rejection rate* (CR) versus *correct acceptance rate* (CA) for different values of the acceptance threshold[a]. The correct rejection rate is the number of incorrectly translated words that have been tagged as wrong, divided by the total number of incorrectly translated words. The correct acceptance rate is the ratio of correctly translated words that have been tagged as correct. These two rates depend on each other: If one of them is restricted by a lower bound, the other one cannot be restricted. ROC curves lie on the unit square and connect the points $(0, 1)$ and $(1, 0)$: random separation of the classes gives a curve that runs along the diagonal, and perfect separation gives one that coincides with the top and right edges of the square. ROC curves thus provide a normalized picture of classifier performance that makes it easy to compare classifiers across the range of (relative) threshold values. Classifier C_1 dominates classifier C_2 if C_1's curve is always above C_2's whenever the two curves differ. It is possible for neither classifier to dominate the other (see section 7.2 for an example of such a curve). Other properties of ROC curves can be found in [Duda et al. 01].

To plot a ROC curve, one typically sorts the words in the test corpus in ascending order by assigned confidence $c(e)$. Then each rank in the resulting corpus corresponds to a distinct pair of (CA,CR) values that can be plotted on a graph. Figure 4.2 gives an example of a sorted data set and the resulting graph. A true class of 1 means that the word is correct, and 0 stands for incorrectness. The first line in the table corresponds to accepting all words. Then, the words are rejected one by one, and the resulting CA and CR are given in the right-most columns of the table.

ROC curves provide for a qualitative analysis of classifier performance; a related quantitative metric is IROC, defined as the area under a ROC curve. From figure 4.2 it is obvious that ROC curves are staircase-shaped: either CA or CR will change with each new point, but never both simultaneously. This makes it easy to calculate the area by simply summing fixed-width columns in the graph. The IROC for the example presented in figure 4.2 is 0.833.

The geometric interpretation makes it obvious that IROC takes on values in $[0, 1]$, with 0.5 corresponding to a random separation of correct and incorrect examples, 1.0 corresponding to a perfect separation (all incorrect examples before correct ones), and 0.0 the opposite. It is less obvious that the baseline value of 0.5 is not affected by the prior probability of correctness in the data set, as is CER, for example. But it can be shown [Blatz et al. 03] that the expected value of the IROC is 0.5. This independence of the prior is a nice property because it means that IROC values from different data sets can be compared directly.

4.3.3 Normalized cross entropy (NCE)

The normalized cross entropy, which was originally suggested by NIST (National Institute of Standards and Technology) [NIST 00], is an evaluation criterion for probabilistic

[a] A variant of the ROC curve is the Detection Error Tradeoff (DET) curve which plots $1 - \text{CR}$ versus $1 - \text{CA}$.

sorted rank	true class	CA	CR
-	-	6/6	0/4
1	0	6/6	1/4
2	0	6/6	2/4
3	1	5/6	2/4
4	0	5/6	3/4
5	1	4/6	3/4
6	1	3/6	3/4
7	0	3/6	4/4
8	1	2/6	4/4
9	1	1/6	4/4
10	1	0/6	4/4

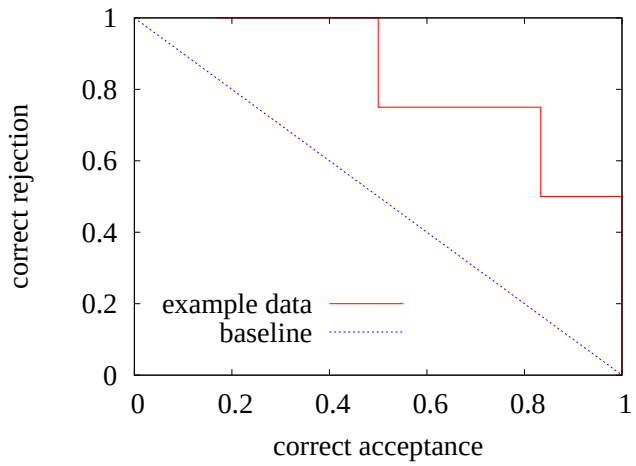


Figure 4.2: Example of a sorted data set and ROC curve.

confidence measures. Like the ROC curves presented above, it measures the performance of the classifier independent of the actual acceptance threshold. It is also independent of the baseline given by the quality of the automatically generated translations. It evaluates how well the word posterior probability models the actual probability of correctness. In other words, the NCE measures the gain in information about the correctness of words obtained with a confidence measure. To this purpose, one resorts to entropy known from information theory [Mathar 96]. The sequence of true classes $c_1^N = c_1, \ldots, c_N$ for the words e_1^N in the corpus is considered as being generated by a random variable X. Here, $c_n = 1$ means that the word is correct, and $c_n = 0$ indicates an incorrect word.

The initial *entropy* $H(X)$ is obtained by assigning each word in the corpus a fixed probability p_0 of correctness:

$$H(X) = -p_0 \cdot \log_2(p_0) - (1 - p_0) \cdot \log_2(1 - p_0) \ .$$

The value for this probability which minimizes the baseline entropy is the prior

$$p_0 = \frac{1}{N} \sum_{n=1}^{N} c_n \ ,$$

which is equal to the accuracy of the MT system, given by the ratio of correct words in the translation. The entropy $H(X)$ expresses the difficulty of determining the correctness of words using only prior information.

The confidence values p_1^N may as well be regarded as a random sequence generated by variable Y. Given these as knowledge source, the *conditional entropy* $H(X|Y)$ denotes the uncertainty which remains when using the confidence measure:

$$H(X|Y) = -\frac{1}{N} \sum_{n=1}^{N} c_n \cdot \log_2 p_n + (1 - c_n) \cdot \log_2(1 - p_n) \ .$$

If the information gain resulting from the confidence estimation is very low, the conditional entropy will be close to the initial one. The difference between the two entropies expresses

the gain obtained by the use of confidence measures. It is known as *cross entropy* or *mutual information*. Since this criterion still depends on the initial entropy of X, i.e. on the quality of the translations which are to be classified, it is divided by $H(X)$ to obtain the *normalized cross entropy*:

$$NCE = \frac{H(X) - H(X|Y)}{H(X)} .$$

If one had perfect knowledge about which words are correct and which are not, one would have $NCE = 1$. If the confidence measure provided no additional information, one would have $NCE = 0$.

In this thesis, the normalized cross entropy is not used because it bears a conceptual disadvantage which makes it impossible to use it for the evaluation of the suggested confidence measures without modifying them. As discussed in section 4.1, the word posterior probabilities defined here can directly be used as a confidence measure. In this case, the normalized cross entropy approaches infinity as soon as the posterior probability of a word equals one, despite the fact that this word has not been translated correctly, as can easily be seen from the equations presented above. Instead of the normalized cross entropy, confidence error rates and ROC curves are presented for the confidence measures proposed in this thesis.

4.4 Combination of confidence features

In related work in MT as well as in speech recognition, the combination of numerous confidence features was suggested [Blatz et al. 04, Gandrabur and Foster 03, Quirk 04, Sanchis 04]. Among the methods used for combination are multi-layer artificial neural networks, naive Bayes classifiers and modified linear regression.

In 2003, one team at the CLSP summer workshop worked on confidence estimation for machine translation. In this workshop, the combination of many different features into one confidence measure was studied. Among them were several of the system-based word posterior probabilities proposed in this thesis. They were calculated over N-best lists provided for the workshop. The investigated variants are the word posterior probabilities

- considering the fixed target position (introduced in section 3.2.1.1),
- regarding the occurrence of the word in any position (see section 3.2.1.5),
- based on the aligned source positions (see section 3.2.2).

For each of these approaches, the word posterior probabilities, the relative frequencies, and the rank-weighted frequencies (as explained in section 3.1) were calculated.

The IBM-1 based confidence measure introduced in equation 3.15 in section 3.3.1 was developed in this workshop. Additionally, the following types of features were studied:

- features based on the underlying SMT model, e.g. the identity of the alignment template that was applied in the translation of the considered target word,

- simple target based features, such as a basic syntax check or the number of occurrences of a word in the sentence,
- semantic features based on WordNet's polysemy count and similar information.

In total, 17 different features were applied for word-level confidence estimation. A description of all features is given in the final report of the workshop [Blatz et al. 03].

The different features were combined in two different ways: using a Naive Bayes classifier [Sanchis 04] and multi-layer perceptrons (MLPs) with different numbers of hidden units [Collobert et al. 02]. The details will be omitted here; they can be found in [Blatz et al. 03]. A short overview of the experimental results on word-level confidence estimation achieved in the workshop will be given in section 7.2.5.

Since the combination of several confidence measures proved successful, the different word posterior probabilities proposed here have been combined with each other. The combination was performed in a log-linear manner. Let $p_m(e \,|\, f_1^J, \ldots)$, $m = 1, \ldots, M$, be the word posterior probabilities of e determined using different approaches. The word confidence resulting from their combination is calculated as

$$c(e) \;=\; \exp\left\{ -\sum_{m=1}^{M} \lambda_m \cdot \log p_m(e \,|\, f_1^J, \ldots) \right\} \;.$$

The interpolation weights λ_m are optimized w.r.t. some confidence evaluation metric on the development corpus using the Downhill Simplex algorithm [Press et al. 02]. With this approach, the confidence error rates have been reduced over the best single confidence measure consistently on all corpora. The experimental results will be presented in section 7.2.5.

However, the focus of this thesis is on word posterior probabilities as stand-alone confidence measures. Their theoretical foundation and the relation between the word posterior probability and the applied word error measure will be studied. Chapter 5 will show that the word posterior probabilities are closely related to the posterior risk.

4.5 Acceptance threshold

The binary classifier defined in equation 4.1 is state of the art [Blatz et al. 03, Sanchis 04, Wessel 02]: The confidence of a target word is compared to a threshold, and the word is classified as correct if the confidence is above this threshold, and rejected otherwise. This threshold is a global one which is determined on the development set. The same threshold is applied for all target words, regardless of the length of the generated target sentence. One might think that this threshold should depend on the length of the hypothesis or the source sentence. The longer the sentence, the more words should be accepted. Therefore, several different methods for classifying words as correct or incorrect were investigated in this thesis. The results will be presented in section 7.2.7.

The classification methods studied in this work are the following:

- Separate thresholds for different sentence lengths:
  One approach pursued is to determine a separate threshold value $t(I)$ for each translation **hypothesis** length I. For each I in the development corpus, this threshold $t(I)$ is optimized w.r.t. CER. Since not all possible sentence lengths will be seen in the development corpus, smoothing over the sentence lengths is applied: the threshold $t(I)$ is determined for all sentences of length $I - w, \ldots, I, \ldots, I + w$ for some window size w.

  When classifying the words in the test corpus, a word in sentence e_1^I is accepted or discarded based on the threshold $t(I)$. If the sentence length I has not been seen in the development corpus, the mean of the thresholds for the nearest two sentence lengths smaller and larger than I is used.

  Since the length I of a generated hypothesis is not necessarily correct, one can also determine a separate threshold $t(J)$ for each **source** sentence length J. This can be expected to be a more reliable estimate of the threshold because it is not affected by incorrect hypothesis length due to translation errors. The smoothing and handling of unseen sentence lengths is the same as described above.

- Threshold as a function of the sentence length:
  Instead of directly optimizing a separate threshold for each source or target sentence length as proposed above, one can also modify the acceptance threshold t based on the sentence length. Let $h(J)$ be a function of the source sentence length J. The resulting classifier is given by

$$\text{class}(e) = \begin{cases} \text{correct} , & \text{if } p(e \mid f_1^J) \geq h(J) \cdot t \\ \text{incorrect} , & \text{otherwise} \end{cases} \tag{4.2}$$

  Possible choices for $h(\cdot)$ include the identity function, the (square) root or the square of J. Analogously, the threshold can be modified by a function of the target hypothesis length I.

- Acceptance of a fixed number of words:
  Instead of setting a threshold on the confidence value of the target words, one can also set a threshold on the number of target words to be accepted for a given source sentence. Assume that I words are to be accepted. Then, all target words are sorted by their confidence in descending order, and the I words with the highest confidence are labeled as correct and all others are discarded. Let $e_{(1)}, \ldots, e_{(V)}$ be all possible target words which are sorted by their confidence in descending order. Then this classifier accepts the words $e_{(1)}, \ldots, e_{(I)}$ and discards $e_{(I+1)}, \ldots, e_{(V)}$:

$$\text{class}(e) = \begin{cases} 1 , & \text{if } p(e \mid f_1^J) \geq p(e_{(I)} \mid f_1^J) \\ 0 , & \text{else} \end{cases} \tag{4.3}$$

  In the experiments performed for this thesis, this number has been varied by the introduction of an offset w, i.e. the $I + w$ target words with the highest confidence are accepted for a given hypothesis of length I. Nevertheless, this assumes that the length of the generated hypothesis is correct in most of the cases. Since this is not necessarily true, it is better to base the decision about the number of accepted

words on the source sentence. A function $h(J)$ determines how many target words should be accepted correct for a given source sentence length J. This function can be developed by examining the training corpora and taking e.g. the ratio of source and target sentence length as indicator.

5 Bayes decision rule and word posterior probabilities

There are three ingredients to any statistical approach to MT, namely the Bayes decision rule, the probability models (trigram language model, HMM, etc., as shown in equation 1.1 in section 1.1) and the training criterion (maximum likelihood, mutual information, ...).

The topic of this chapter is to examine the relation between Bayes decision rule and word posterior probabilities as defined in chapter 3. To this end, the differences between string error (or *sentence* error) and symbol error (or *word* error) and their implications for the Bayes decision rule are investigated. For several different word error measures, the so-called loss function will be formulated, and the posterior risk will be derived. As will be shown, this is closely related to the word posterior probabilities presented in section 3.2. This will yield a sound theoretical foundation of the word posterior probabilities and their use as confidence measures.

5.1 The Bayes posterior risk

Knowing that any task in natural language processing (NLP) is a difficult one, the goal is to keep the number of wrong decisions as small as possible. To classify an observation vector y into one out of several classes c, one resorts to statistical decision theory and tries to minimize the **posterior risk** $R(c|y)$ in taking a decision. This risk is defined as

$$R(c|y) = \sum_{\tilde{c}} Pr(\tilde{c}|y) \cdot L[c, \tilde{c}] \ , \tag{5.1}$$

where $L[c, \tilde{c}]$ is the so-called **loss function** or **error measure**. This expresses the loss which is incurred in making decision $\tilde{c}$ when the true class is c. The resulting decision rule is known as **Bayes decision rule** [Duda et al. 01]:

$$y \rightarrow \hat{c} \ = \ \underset{c}{\operatorname{argmin}} \, R(c|y) \ = \ \underset{c}{\operatorname{argmin}} \left\{ \sum_{\tilde{c}} Pr(\tilde{c}|y) \cdot L[c, \tilde{c}] \right\} \ . \tag{5.2}$$

In the following, two specific forms of the error measure, $L[c, \tilde{c}]$, will be considered. The first will be the measure for *sentence* errors, which is the typical loss function used in virtually all statistical approaches to machine translation. The second is the measure for *word* errors, which is more appropriate for machine translation. There exist several ways of determining translation errors on the word level (see section 4.2). Three out of those will be considered in the following. The Bayes decision rule for these word error measures

will be formulated. From this, the word posterior probabilities introduced in section 3.2 can be derived.

5.2 Sentence error

For machine translation, the starting point is the observed sequence of words $y = f_1^J = f_1...f_J$, i.e. the sequence of words in the source language which has to be translated into a target language sequence $c = e_1^I = e_1...e_I$.

The first error measure considered here is the sentence error: two target language sentences are said to be identical only when the words in each position are identical (which naturally requires the same length I). In this case, the error measure L between two strings e_1^I and $\tilde{e}_1^{\tilde{I}}$ is:

$$L[e_1^I, \tilde{e}_1^{\tilde{I}}] \;=\; 1 - \delta(I, \tilde{I}) \cdot \prod_{i=1}^{I} \delta(e_i, \tilde{e}_i) \;,$$

with the Kronecker delta $\delta(\cdot, \cdot)$. In other words, the errors are counted at the *string* (or sentence) level and not at the level of single symbols (or words). Inserting this cost function into the Bayes risk (see equation 5.1), the following form of *Bayes decision rule for minimum sentence error* is obtained:

$$\begin{aligned} f_1^J \rightarrow (\hat{I}, \hat{e}_1^{\hat{I}}) \;&=\; \operatorname*{argmax}_{I, e_1^I} \left\{ Pr(I, e_1^I | f_1^J) \right\} \\ &=\; \operatorname*{argmax}_{I, e_1^I} \left\{ Pr(I, e_1^I, f_1^J) \right\} \;. \end{aligned}$$

This is the starting point for virtually all statistical approaches in machine translation. However, this decision rule is only optimal when *sentence* error is considered. In practice, the empirical errors are counted at the *word* level. This inconsistency of decision rule and error measure is rarely addressed in the literature.

5.3 Word error

Instead of using the *sentence* error rate, the errors can also be counted on the level of *symbols* or single *words*. In the MT research community, there exist several different error measures that are based on the word error (see section 4.2 and appendix B). In relation to Bayes decision rule, the word error rate (WER), the position independent word error rate (PER), and the error measure Pos will be investigated in the following.

For NLP tasks where there is no variance in the string length (such as Part-of-Speech tagging), the integration of the symbol error measure into Bayes decision rule yields that a maximization of the posterior probability for each position i has to be performed [Ney et al. 04]. In machine translation, a method for accounting for differences in sentence length or word order between the two regarded strings is needed. One such method is the Levenshtein alignment which is performed in WER calculation.

The presentation will start with the most rigorous word error metric Pos which considers only those words as correct which occur in exactly the same position as they do in the reference. Then, this constraint will be relaxed by the introduction of the Levenshtein alignment, resulting in the widely used word error metric WER. Afterwards, the posterior risk will be derived for PER which does not consider the word position at all, but only the frequency of the word in the sentence.

5.3.1 Exact target position (Pos)

If Pos is applied as word error measure, only those words are considered as correct which occur exactly in the target position given in the reference. This yields a rather strict loss function which does not allow for variation in the sentence position.

Assume first that the sentences to be compared, e_1^I and $\tilde{e}_1^I$, have the same length I. Then the loss function L has the following form:

$$L[e_1^I, \tilde{e}_1^I] \;=\; \sum_{i=1}^{I} [1 - \delta(e_i, \tilde{e}_i)] \;\; .$$

The integration of this loss function into the posterior risk yields:

$$\begin{aligned}
R(e_1^I|f_1^J) &= \sum_{\tilde{e}_1^I} Pr(\tilde{e}_1^I|f_1^J) \cdot L[e_1^I, \tilde{e}_1^I] \\
&= \sum_{\tilde{e}_1^I} Pr(\tilde{e}_1^I|f_1^J) \cdot \sum_{i=1}^{I} [1 - \delta(e_i, \tilde{e}_i)] \\
&= \sum_{i=1}^{I} \sum_{\tilde{e}_1^I} Pr(\tilde{e}_1^I|f_1^J) - \sum_{i=1}^{I} \sum_{\tilde{e}_1^I} \delta(e_i, \tilde{e}_i) \cdot Pr(\tilde{e}_1^I|f_1^J) \\
&= I - \sum_{i=1}^{I} \sum_{\tilde{e}_1^I} \delta(\tilde{e}_i, e_i) \cdot Pr(\tilde{e}_1^I|f_1^J) \;\; ,
\end{aligned}$$

where $\sum_{\tilde{e}_1^I} \delta(\tilde{e}_i, e_i) \cdot Pr(\tilde{e}_1^I|f_1^J)$ is the posterior probability of word e to occur exactly in position i of the target sentence. This corresponds exactly to the word posterior probability defined in equations 3.1 and 3.2 on page 14. So this value is directly related to the posterior risk for the error measure Pos. Applying these word posterior probabilities as confidence measures for classification w.r.t. Pos should naturally yield the best result which can be obtained. The posterior risk for Pos yields the decision rule:

$$f_1^J \rightarrow \tilde{e}_1^I \;=\; \operatorname*{argmin}_{e_1^I} \left\{ I - \sum_{i=1}^{I} \sum_{\tilde{e}_1^I} \delta(\tilde{e}_i, e_i) \cdot Pr(\tilde{e}_1^I|f_1^J) \right\} \;\; .$$

This means that the optimal translation result w.r.t. Pos can be achieved by minimizing the value of the sentence length I minus the sum of the word posterior probabilities. Since

I is fixed in this setting, this decision results in maximizing the posterior probabilities of all words in the sentence. So the Bayes decision rule can be rewritten as

$$f_1^J \rightarrow \tilde{e}_1^I \;=\; \operatorname*{argmax}_{e_1^I} \left\{ \sum_{i=1}^{I} \sum_{\tilde{e}_1^I} \delta(\tilde{e}_i, e_i) \cdot Pr(\tilde{e}_1^I | f_1^J) \right\} \;.$$

This is exactly the situation described at the end of section 5.1: If the sentence length is given, the optimal result is achieved by maximizing the sum of the word posterior probabilities. Unfortunately, this is not the case in MT, so the loss function has to be modified to account for sentences of different lengths.

Consider a generalized error measure which compares sentences e_1^I and $\tilde{e}_1^{\tilde{I}}$ of different lengths. The resulting loss function is the following:

$$L[e_1^I, \tilde{e}_1^{\tilde{I}}] \;=\; \max\{\tilde{I} - I, 0\} + \sum_{i=1}^{I} [1 - \delta(e_i, \tilde{e}_i)] \;,$$

where $\delta(e_i, \tilde{e}_i) := 0$ is defined for $i > \tilde{I}$. The integration of this loss function into the posterior risk yields:

$$\begin{aligned}
&R(e_1^I | f_1^J) = \\
&= \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} Pr(\tilde{e}_1^{\tilde{I}} | f_1^J) \cdot L[e_1^I, \tilde{e}_1^{\tilde{I}}] \\
&= \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} Pr(\tilde{e}_1^{\tilde{I}} | f_1^J) \cdot \left(\max\{\tilde{I} - I, 0\} + \sum_{i=1}^{I} [1 - \delta(e_i, \tilde{e}_i)] \right) \\
&= \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} Pr(\tilde{e}_1^{\tilde{I}} | f_1^J) \cdot \max\{\tilde{I} - I, 0\} + \sum_{i=1}^{I} \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} Pr(\tilde{e}_1^{\tilde{I}} | f_1^J) - \sum_{i=1}^{I} \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} \delta(e_i, \tilde{e}_i) \cdot Pr(\tilde{e}_1^{\tilde{I}} | f_1^J) \\
&= \sum_{\tilde{I} > I} Pr(\tilde{I} | f_1^J) \cdot (\tilde{I} - I) + I - \sum_{i=1}^{I} \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} \delta(\tilde{e}_i, e_i) \cdot Pr(\tilde{e}_1^{\tilde{I}} | f_1^J) \;,
\end{aligned}$$

where $Pr(I | f_1^J)$ is the probability of seeing a target sentence of length I as a translation of the source sentence f_1^J. This probability can be obtained by summing over all possible target sentences of length I, i.e.

$$Pr(I | f_1^J) \;=\; \sum_{\tilde{I}, \tilde{e}_1^{\tilde{I}}} \delta(\tilde{I}, I) \cdot Pr(\tilde{e}_1^{\tilde{I}} | f_1^J) \;.$$

This yields the Bayes decision rule

$$f_1^J \rightarrow \tilde{e}_1^{\tilde{I}} \;=\; \operatorname*{argmin}_{I, e_1^I} \left\{ I + \sum_{\tilde{I} > I} Pr(\tilde{I} | f_1^J) \cdot (\tilde{I} - I) - \sum_{i=1}^{I} \sum_{\tilde{e}_1^{\tilde{I}}} \delta(\tilde{e}_i, e_i) \cdot Pr(\tilde{e}_1^{\tilde{I}} | f_1^J) \right\} \;.$$

Thus, if different sentence lengths are possible for e_1^I, the Bayes decision rule has to consider the probability distribution over the sentence lengths as well. There is still a dependency on the word posterior probabilities defined in section 3.2.1.1, but now the sentence length I has to be considered as well in the minimization operation.

5.3.2 WER

In the following, the Bayes decision rule for the word error rate (WER), which is widely used in MT evaluation, will be derived. For two sentences e_1^I and $\tilde{e}_1^{\tilde{I}}$, the Levenshtein alignment is denoted by $\mathcal{L} = \mathcal{L}(e_1^I, \tilde{e}_1^{\tilde{I}})$; for a word e_i the Levenshtein aligned word in $\tilde{e}_1^{\tilde{I}}$ is denoted by $\mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}})$ for $i = 1, \ldots, I$. If e_i is an insertion, then $\mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}}) = e_0$, the so-called empty word. Let $d(e_1^I, \tilde{e}_1^{\tilde{I}})$ be the number of deleted words in $\tilde{e}_1^{\tilde{I}}$ according to the Levenshtein alignment.

Again a simplified case will be considered first: Assume that the two sentences under comparison, e_1^I and $\tilde{e}_1^I$, are of the same length I. For this special case of the WER, the number of insertions is the same as the number of deletions. Thus, the deletions can also be expressed by the number of insertions which will be useful when determining the posterior risk. The simplified WER is defined by the loss function:

$$\begin{aligned} L[e_1^I, \tilde{e}_1^I] &= d(e_1^I, \tilde{e}_1^I) + \sum_{i=1}^{I} \left[1 - \delta(e_i, \mathcal{L}_i(e_1^I, \tilde{e}_1^I))\right] \\ &= \sum_{i=1}^{I} \delta(e_0, \mathcal{L}_i(e_1^I, \tilde{e}_1^I)) + \sum_{i=1}^{I} \left[1 - \delta(e_i, \mathcal{L}_i(e_1^I, \tilde{e}_1^I))\right] \ . \end{aligned}$$

The posterior risk is then given by

$$\begin{aligned} &R(e_1^I | f_1^J, I) = \\ &= \sum_{\tilde{e}_1^I} L[e_1^I, \tilde{e}_1^I] \cdot Pr(\tilde{e}_1^I | f_1^J) \\ &= \sum_{\tilde{e}_1^I} Pr(\tilde{e}_1^I | f_1^J) \cdot \sum_{i=1}^{I} \delta(e_0, \mathcal{L}_i(e_1^I, \tilde{e}_1^I)) + \sum_{\tilde{e}_1^I} Pr(\tilde{e}_1^I | f_1^J) \cdot \sum_{i=1}^{I} \left[1 - \delta(e_i, \mathcal{L}_i(e_1^I, \tilde{e}_1^I))\right] \\ &= \sum_{i=1}^{I} \sum_{\tilde{e}_1^I} \delta(\mathcal{L}_i(e_1^I, \tilde{e}_1^I), e_0) \cdot Pr(\tilde{e}_1^I | f_1^J) + I - \sum_{i=1}^{I} \sum_{\tilde{e}_1^I} \delta(\mathcal{L}_i(e_1^I, \tilde{e}_1^I), e_i) \cdot Pr(\tilde{e}_1^I | f_1^J) \ . \end{aligned}$$

Since the length I is fixed, it can be omitted in the Bayes decision rule, which yields:

$$\begin{aligned} &f_1^J \rightarrow \tilde{e}_1^I = \\ &= \operatorname*{argmin}_{e_1^I} \left\{ \sum_{i=1}^{I} \left(\sum_{\tilde{e}_1^I} \delta(\mathcal{L}_i(e_1^I, \tilde{e}_1^I), e_0) \cdot Pr(\tilde{e}_1^I | f_1^J) - \sum_{\tilde{e}_1^I} \delta(\mathcal{L}_i(e_1^I, \tilde{e}_1^I), e_i) \cdot Pr(\tilde{e}_1^I | f_1^J) \right) \right\} \ . \end{aligned}$$

The probabilities which are used in the second sum are the word posterior probabilities defined in section 3.2.1.2. So the word posterior probabilities are closely related to the loss function for WER. They can thus be expected to perform well as confidence measures if the reference tags are given by WER.

Now consider an extended formulation of the loss function presented above, accounting for sentences of different lengths as well. This is then the actual loss for the WER:

$$L[e_1^I, \tilde{e}_1^{\tilde{I}}] = d(e_1^I, \tilde{e}_1^{\tilde{I}}) + \sum_{i=1}^{I} \left[1 - \delta(e_i, \mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}}))\right] \ .$$

Inserting this loss function into the posterior risk yields:

$$\begin{aligned} R(e_1^I|f_1^J) &= \sum_{\tilde{I},\tilde{e}_1^{\tilde{I}}} Pr(\tilde{e}_1^{\tilde{I}}|f_1^J) \cdot L[e_1^I, \tilde{e}_1^{\tilde{I}}] \\ &= \sum_{\tilde{I},\tilde{e}_1^{\tilde{I}}} Pr(\tilde{e}_1^{\tilde{I}}|f_1^J) \cdot d(e_1^I, \tilde{e}_1^{\tilde{I}}) + I - \sum_{i=1}^{I} \sum_{\tilde{I},\tilde{e}_1^{\tilde{I}}} \delta(\mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}}), e_i) \cdot Pr(\tilde{e}_1^{\tilde{I}}|f_1^J) \ . \end{aligned}$$

Now the decision cannot be taken by considering only the word posterior probabilities any longer. The posterior risk includes an additional term which depends on the actual word sequence because the deletions in the Levenshtein alignment of the whole sequence are explicitly taken into account.

5.3.3 PER

Unlike the WER, the PER compares the words in the two sentences without taking the word order into account. It only considers the number of times a word occurs in a sentence, but not its position. The loss function for the PER can be expressed by considering the counts of each word in the sentence. Let n_e be the count of word e in sentence e_1^I, and $\tilde{n}_e$ its frequency in $\tilde{e}_1^{\tilde{I}}$, respectively. Then, $min(n_e, \tilde{n}_e)$ occurrences of the word are correct and do not contribute to the PER loss. Consider the sentence e_1^I: if it is longer than $\tilde{e}_1^{\tilde{I}}$, then the PER is determined by $I - \sum_e \min(n_e, \tilde{n}_e)$ substitution and insertion operations. If it is shorter, $\tilde{I} - \sum_e \min(n_e, \tilde{n}_e)$ substitution and deletion operations have to be performed. That is, the PER loss can be expressed as

$$\begin{aligned} L[e_1^I, \tilde{e}_1^{\tilde{I}}] &= \max(I, \tilde{I}) - \sum_e \min(n_e, \tilde{n}_e) \\ &= \frac{1}{2}\left(I + \tilde{I} + |I - \tilde{I}|\right) - \frac{1}{2}\sum_e \left(n_e + \tilde{n}_e - |n_e - \tilde{n}_e|\right) \\ &= \frac{1}{2}\left(I + \tilde{I} + |I - \tilde{I}| - \sum_e n_e - \sum_e \tilde{n}_e + \sum_e |n_e - \tilde{n}_e|\right) \\ &= \frac{1}{2}\left(|I - \tilde{I}| + \sum_e |n_e - \tilde{n}_e|\right) \ . \end{aligned}$$

This representation does not depend on the target word sequences e_1^I and $\tilde{e}_1^{\tilde{I}}$, but only on the counts of the words. Let $n_1^E = n_1, \ldots, n_E$ and $\tilde{n}_1^E$ be the count sequences for all words $e = 1, \ldots, E$ in the vocabulary. Note that $I = \sum_e n_E$ and $\tilde{I} = \sum_e \tilde{n}_E$. Then, the loss can be expressed as the loss of the two count sequences:

$$L[n_1^E, \tilde{n}_1^E] = \frac{1}{2}\left(|I - \tilde{I}| + \sum_e |n_e - \tilde{n}_e|\right) \ .$$

The distribution of the counts, $Pr(n_1^E|f_1^J)$, can be obtained by summing up the probabilities of all sentences $\tilde{e}_1^I$ for which the word counts are equal to n_1^E, i.e.

$$Pr(n_1^E|f_1^J) = \sum_{\tilde{e}_1^I} \delta(\tilde{n}_1^E, n_1^E) \cdot Pr(\tilde{e}_1^{\tilde{I}}|f_1^J) \ .$$

The integration of this loss function into the posterior risk yields

$$\begin{aligned} R(n_1^E|f_1^J) &= \sum_{\tilde{n}_1^E} Pr(\tilde{n}_1^E|f_1^J) \cdot L[n_1^E, \tilde{n}_1^E] \qquad (5.3) \\ &= \sum_{\tilde{n}_1^E} Pr(\tilde{n}_1^E|f_1^J) \cdot \left(\frac{1}{2}|I - \tilde{I}| + \frac{1}{2} \sum_e |n_e - \tilde{n}_e| \right) \\ &= \frac{1}{2} \sum_{\tilde{n}_1^E} |I - \tilde{I}| \cdot Pr(\tilde{n}_1^E|f_1^J) + \frac{1}{2} \sum_e \sum_{\tilde{n}_1^E} |n_e - \tilde{n}_e| \cdot Pr(\tilde{n}_1^E|f_1^J) \\ &= \frac{1}{2} \sum_{\tilde{I}} |I - \tilde{I}| \cdot Pr(\tilde{I}|f_1^J) + \frac{1}{2} \sum_e \sum_{\tilde{n}_e} |n_e - \tilde{n}_e| \cdot Pr_e(\tilde{n}_e|f_1^J) \ , \end{aligned}$$

where $Pr_e(n_e|f_1^J)$ is the posterior probability of the count n_e of word e. It can be computed by summing up the probabilities of all sentences containing target word e (exactly) n_e times as described in section 3.2.3. So this formulation of the Bayes decision rule for PER yields a theoretical foundation for the count-based word posterior probabilities introduced in section 3.2.3. As the experimental results presented in section 7.2.1 will show, these word posterior probabilities are one of the best confidence measures for PER based classification.

6 Application of word confidence measures in interactive SMT

The work presented in this chapter deals with the application of confidence estimation in an interactive machine translation system. Interactive machine translation systems aim at improving the productivity of human translators. They suggest translations of the source text and take text into account that the user has already typed. The prototype has been developed in the joint European project TransType2 [tt2 05]. The goal of this project was to combine two approaches: computer-aided translation, where a machine translation system suggests translations which can be modified by a human user, and a state-of-the-art statistical approach to machine translation. Before, only rather simple translation models had been integrated into such systems, e.g. in the Canadian TransType project [Foster et al. 02].

Confidence estimation has been integrated into the interactive system in order to improve the quality of translations predicted by the system. Since the goal is to reduce user effort, one has to consider the gain in keystrokes needed to type the translation as well as the time that he or she spends on reading and deciding whether to accept a suggestion. That is, the system has to keep the balance between the benefit of long predictions and the negative effect of incorrect predictions. Confidence estimation is applied as a way to achieve this balance.

The system uses confidence estimation in two different ways: for the *selection* of words in the extension (as described in section 6.2.1) and for the *rejection* of words with low confidence (see section 6.2.2). These methods improve the prediction accuracy of the interactive system. This yields a reduction in typing effort of a human as the results presented in section 7.4 will show. The system performance can be further increased by taking a correct prefix entered by the user into account for confidence estimation. The confidence measures can be modified to account only for untranslated words in the source sentence. This will be the topic of section 6.2.3.

6.1 Interactive statistical machine translation

In the setting investigated here, a state-of-the-art SMT system, namely the alignment templates system described in section C.1, is employed in an interactive translation environment in the following way: Given the source sentence, the system proposes a translation. A human translator checks this translation from left to right, correcting the first error. The SMT system then proposes a new extension, taking the correct prefix $e_1^i = e_1 \dots e_i$ into account. These steps are repeated until the whole input sentence has

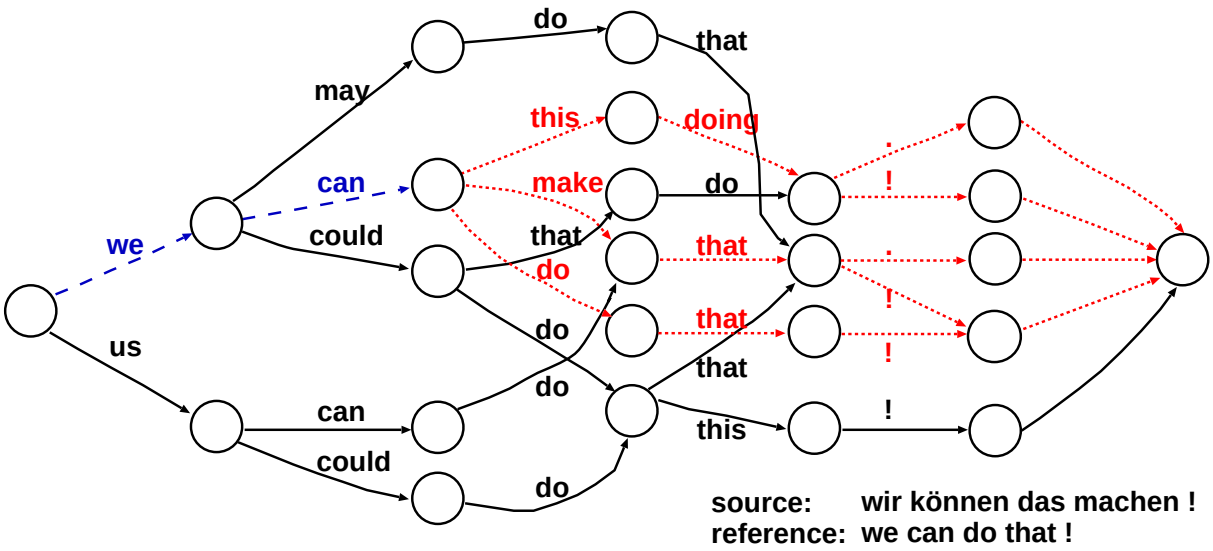


Figure 6.1: Example of search in interactive SMT on a word graph. The given prefix "we can" is marked by dashed blue lines, possible extensions in dotted red.

been correctly translated. The decision rule introduced in equation 1.1 on page 2 is then modified so that the maximization is performed over all possible extensions e_{i+1}^I of the prefix e_1^i:

$$\hat{e}_{i+1}^{\hat{I}} \quad = \quad \underset{I, e_{i+1}^I}{\operatorname{argmax}} \left\{ Pr(e_{i+1}^I \,|\, e_1^i,\, f_1^J) \right\} \ .$$

For reasons of simplicity, this equation is formulated on the word level. In the actual implementation, the same method is applied on the character level, and the search for the extension is performed after each keystroke of the human translator. This requires a highly efficient search, because human users will only accept response times of fractions of a second. To achieve this, the SMT system computes a word graph representing the most likely translations of the input sentence [Ueffing et al. 02]. This representation of the search space is then used for the efficient computation of the extension. An example of such a word graph is given in figure 6.1. Note that this graph is much smaller than the ones which are actually employed in the system. Assume that the prefix "we can" (marked by dashed blue lines in the figure) is given, and the system searches for possible extensions. It will then consider all paths marked by dotted red lines and choose the one with maximal probability. If the given prefix is not found in the word graph, the path with minimal Levenshtein distance to the prefix is selected. For a detailed description of the interactive SMT system, see [Och et al. 03].

6.2 Application of confidence measures

The procedure described in the last subsection can be enhanced through the use of confidence measures. There exist several points in the interactive search algorithm where

they can be applied. When the human user has entered a character or accepted part of the proposed translation, the interactive SMT system starts searching for an appropriate extension of this prefix. It performs the following steps (see [Och et al. 03]):

1. locate the prefix in the word graph,
2. search for the best extension in the word graph,
3. if no good extension is found: use language model only for prediction .

Word confidence measures have been integrated into the interactive SMT system in steps 2 and 3. Different ways of incorporating them have been investigated; they will be described in sections 6.2.1 through 6.2.3.

For the application in an interactive environment, confidence measures are needed that operate on the word level (instead of the sentence level) and that can be computed very efficiently. In the system described here, the IBM-1 based confidence measure introduced in equation 3.16 in section 3.3.1 has been implemented. It determines the maximal lexicon probability of a target word over the source sentence. This measure has been chosen because it relies only on the source sentence and the proposed extension and not on an N-best list or additional information as many other word confidence measures do. Thus, it can be calculated easily and very fast during search. Moreover, its performance in identifying correct words is quite promising given its simplicity as the results presented in section 7.2 will show.

6.2.1 Selection of words

One way of incorporating confidence measures into the interactive system is to choose the next translation prediction according to its confidence. This approach is pursued in [Gandrabur and Foster 03] and the authors report significant improvements in performance. The system investigated there is a rather simple system combining a trigram language model and the IBM translation model 2 [Brown et al. 93]. Phrases of up to four words are predicted based on their confidence value (which is calculated for the whole phrase). The alignment templates system applied in this thesis is more sophisticated than the one investigated by [Gandrabur and Foster 03]. It uses state-of-the-art SMT models. The whole sentence instead of a short phrase is predicted at once. The confidence is estimated for each single word in this prediction. Since the underlying SMT system is more sophisticated than the one used by Gandrabur and Foster, the gain from the confidence based selection can be expected to be lower for the here.

The confidence based prediction is applied in the interactive SMT system in search steps 2 and 3 explained above. The word confidence and the original score assigned by the SMT system are combined in a log-linear manner (with the scaling factor λ):

$$s(e) \quad = \quad \exp\{-\lambda \cdot \log \max_j p(e_i|f_j) - (1-\lambda) \cdot \log p_{\text{base}}(e_i|f_1^J, e_1^{i-1})\} \ , \qquad (6.1)$$

where $\max_j p(e_i|f_j)$ is the word confidence, and $p_{\text{base}}(e_i|f_1^J, e_1^{i-1})$ is the probability of the word as assigned by the base SMT model. Thus, the word confidence is considered

as additional knowledge source when searching for the next extension. This is especially useful in search step 3 described above, because the baseline system predicts the extensions based only on their language model score.

6.2.2 Rejection of words

A novel way of incorporating confidence estimation into the interactive system is to reject proposed words if their confidence is below a given threshold. The SMT system calculates the confidence of each word in the possible extensions. For the words contained in the word graph, this calculation can be done already when the word graph is constructed.

When searching the word graph for the best extension as described in step 2 above, the system discards all completions beginning with a word with low confidence. Among all confident words, it will propose the one with maximal base model probability. That is, the space of possible translations is restricted by considering only highly confident words. If at a certain point all words that are possible extensions are rejected, the system will stop predicting translations. Thus, the system does not necessarily propose whole-sentence extensions anymore. This approach aims at preventing the prediction of long, incorrect extensions. Once the user has entered another character, the interactive system starts again to search for extensions and to propose them.

The search on the word graph can be unsuccessful if the given prefix (or a similar string[a]) is not found in the graph or if all possible extensions are discarded due to low confidence. Then, the system will predict single word extensions based on the language model only (see step 3 in section 6.2). Again, the confidence module rejects words with low confidence as follows: The possible extensions proposed by the system are sorted by their language model probability given the prefix as history. The interactive system selects the word which is highest in this list and for which the confidence is above the threshold. If none of those words is confident, the system proposes the one with the highest language model score.

Figure 6.2 shows the example word graph introduced earlier after the application of confidence measures for rejection. Assume that the words “doing” and “make” which are on paths representing possible extensions have low confidence and are thus rejected. If the system reaches these words in the graph, and there are no other possible extensions left, it will stop predicting. In figure 6.2, this would be the case if the system extended the prefix “we can” by “this”: The only path completing this sentence starts with an unconfident word. The system would thus stop predicting in order not to annoy a human user through bad propositions.

6.2.3 Use of prefix information

The confidence estimation described so far does not take into account that in the interactive use, the user has already accepted and/or typed a part of the translation. A way of exploiting this knowledge will be introduced in this section.

[a] with respect to Levenshtein distance on character level

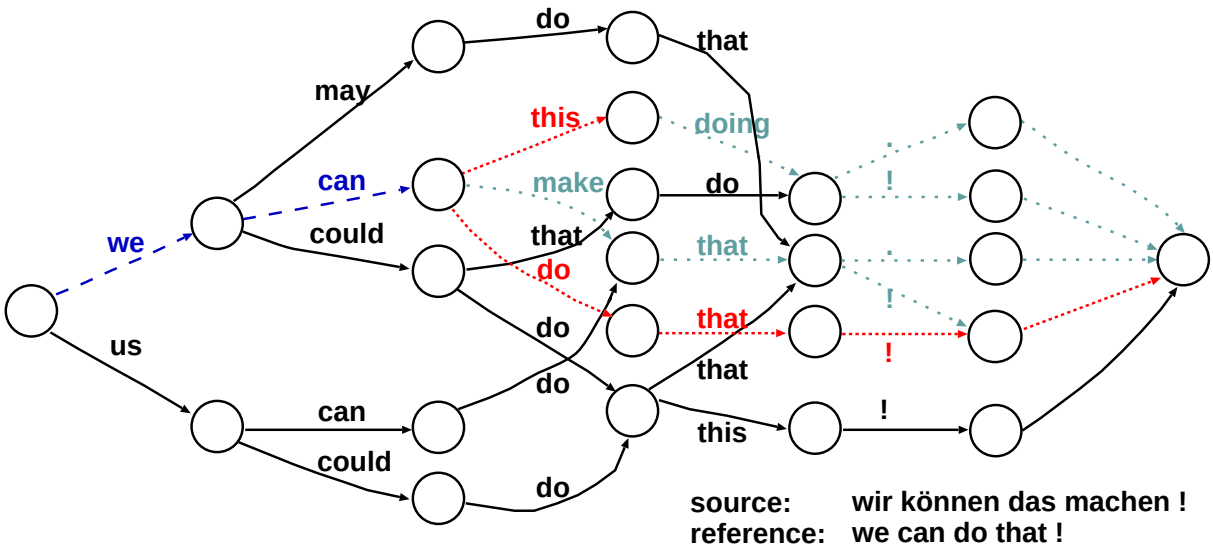


Figure 6.2: Example of search in interactive SMT on a word graph. The given prefix "we can" is marked by dashed blue lines. Possible extensions which have been rejected are marked in thinly dotted grey-blue, those with sufficiently high confidence in dotted red.

Since the confidence of target word e is calculated as the maximal lexicon probability over the source sentence, this calculation can be restricted to those source words that are not covered by the given prefix. An example of this will be given in section 7.4.3. If a prefix word has no correspondence in the source sentence, this will not influence the confidence estimation. Assume a given prefix e_1^i covers the source words $F(e_1^i)$. The confidence $c(e)$ of target word e in a prediction is then given by

$$c(e) \quad = \quad \max_{f_j \notin F(e_1^i)} p(e \,|\, f_j) \ . \tag{6.2}$$

This prevents the system from proposing translations of source words that have already been covered by the prefix.

Another way of using the knowledge contained in the prefix is the adaptation of the confidence threshold. Since the prefix words are known to be correct, it is assumed that they should be accepted by the confidence module. Different source sentences might have different inherent translation difficulties yielding different confidence values. To account for this fact, the confidence of each word in the prefix is compared to the confidence threshold which is lowered if necessary. In order not to adapt the threshold to outliers, it is never lowered by more than half its value. That is, the adapted confidence threshold t_{adapt} for a given prefix e_1^i is determined as

$$t_{\text{adapt}} = \max\{0.5 \cdot t \ , \ \min\{t \, , \ \min_{1 \le k \le i} p_{\text{maxlex}}(e_k \,|\, f_1^J)\} \ \} \ . \tag{6.3}$$

6.3 Evaluation

This subsection describes how an interactive MT system can be evaluated automatically. The existing evaluation measure KSR is discussed, and an extension to this measure is motivated and presented.

6.3.1 Simulated interactive mode

In this subsection, the off-line simulation of interactive statistical machine translation will be described. The proposed extensions are compared against a given reference translation. At first, the system proposes a translation of the source sentence. This is compared to the reference from left to right, and the matching part is accepted. The next character from the reference is appended to the correct prefix, simulating the human user who types something. The system then updates the proposed translation. These steps are repeated until the whole reference has been generated. This mode reflects the application, where the system attempts to match what a human user has in mind, and not simply to produce any correct translation.

A simplified example is shown in figure 6.3. It illustrates the interaction between the system and a simulated user. In practice, the system should translate this short sentence correctly without any user interaction. The reference translation is "what did you say ?" and the first suggestion of the system is "what do you say ?". So, the user accepts the prefix "what d" with one keystroke (denoted with a "#") and then enters the correct character "i". The next suggestion of the system is "what did you said ?". Now, the user accepts the prefix "what did you sa" and then types the character "y". Finally, the system suggests the correct translation, and the user simply accepts by typing "#".

step no.	source	was hast du gesagt ?
	reference	what did you say ?
1	prefix	
	extension	what do you say ?
	user	#i
2	prefix	what di
	extension	d you said ?
	user	#y
3	prefix	what did you say
	extension	?
	user	#

Figure 6.3: Example of simulated user–system interaction in interactive statistical machine translation.

6.3.2 Evaluation measures

In the experiments reported on interactive statistical machine translation so far [Gandrabur and Foster 03, Och et al. 03, Civera et al. 04], the evaluation was based on the number of keystrokes or the time saved by a user when typing the reference translation. Here, this is accounted for by measuring the precision and the recall of the predictions. The recall can be measured by the so-called **keystroke ratio (KSR)** introduced in [Och et al. 03]. It divides the number of keystrokes needed to produce the single reference translation using the interactive translation system by the number of keystrokes needed to type the reference translation:

$$\text{KSR} = \frac{\#\text{keystrokes needed to type the reference } \textit{with} \text{ completion}}{\#\text{keystrokes needed to type the reference } \textit{without} \text{ completion}}$$

The simplifying assumption is made that the user can accept an arbitrary length of the proposed extension using a single keystroke. A keystroke ratio of 1 means that the system is never able to suggest a correct extension. This value gives an indication about the possible effective gain that can be achieved if the interactive translation system is used in a real translation task. On the one hand, the keystroke ratio is very optimistic with respect to the efficiency gain of the user. On the other hand, it is a well-defined objective criterion.

The KSR has the shortcoming that it does not penalize long predictions of bad quality, e.g. the prediction of 10 incorrect words results in the same KSR as the prediction of one incorrect word. It can be seen as a metric measuring **recall** error on the proposed characters versus the reference. To overcome this problem, an additional measure is used in this thesis which determines the ratio of characters in the proposed extension that are correct. By doing so, the **precision** of the predictions is evaluated as well. This accounts also for the reading time that a user spends on predictions even if he or she does not accept them. The precision is measured by determining the Levenshtein alignment between the proposed extension and the (uncovered part of the) reference on word and character basis. All characters which are aligned to themselves are counted as correct, and all others as incorrect. The number of correct characters is divided by the total number of characters in the predicted extension(s).

The combination of precision and recall using the harmonic mean yields the **F-measure** for prediction quality

$$F_p = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} ,$$

where $Recall = 1 - KSR$. This evaluation measure lies in the interval between 0 and 1 and measures quality. It will penalize a system that proposes very short extensions as well as one that proposes long translations of bad quality.

As additional performance measure, the **number of extensions** proposed by the system, i.e. the number of user–system interactions, is counted. Every time the system proposes an extension, the user has to read it and to decide whether to accept or not. Thus, a high number of user–system interactions significantly increases the cognitive load of the user. Furthermore, it shows that the quality of predictions is low, because the user

often discards them. Note that this measure is different from the number of keystrokes the user has to type: If the user accepts part of the proposed completion and types an additional character, this counts as two keystrokes (see the example in figure 6.3). However, if the user does not accept anything from the completion and simply types the next character, this yields only one additional keystroke. The number of completions on the other hand weights each interaction between user and system equally.

In the example given in figure 6.3, the simulated user needs 5 keystrokes to produce the reference translation with the interactive translation system. Simply typing the reference translation would take 19 keystrokes (including blanks and a return at the end). So, the keystroke ratio is $5/19 = 26.3\%$. In total, 31 characters are proposed by the system, out of which 3 are incorrect. The precision of the proposed extensions thus is $28/31 = 90.3\%$. Combining precision and recall yields an F_p value of 81.2% for this small example. The number of user–system interactions is 3.

Experimental results on the application of confidence estimation in the interactive alignment templates system will be presented in section 7.4.

7 Experimental results

This chapter reports on the experimental results for all confidence measures described in chapter 3. First, the experimental setting will be described in section 7.1. Then in section 7.2, the classification performance of the different confidence measures will be presented. Each measure will be analyzed in detail on data from the TransType2 project (TT2). Additionally, a comparison of the different approaches will be given on these data and on the EPPS collection and on data from the NIST MT evaluation campaign. Moreover, alternative ways of defining the classifier will be investigated. A short excursion will describe experiments conducted in the CLSP summer workshop 2003. Section 7.3 will report on rescoring experiments using confidence measures. Experimental results of an interactive machine translation system with confidence estimation will be shown in section 7.4.

7.1 Experimental setting

The experiments have been performed on several different language pairs: French-English, Spanish-English, German-English, and Chinese-English. The corpora have been compiled in the EU projects TransType2 and TC-STAR and for the NIST MT evaluation campaign. The TransType2 corpora consist of technical manuals for Xerox devices such as printers. This domain is very specialized with respect to terminology and style. The corpus statistics are given in tables A.1 through A.3 in appendix A. Most of the detailed investigations have been carried out on these data. The corpus has the advantage of being relatively small. This makes it possible to run many comparative experiments and time or memory consuming computations. The methods which proved best on this task have additionally been studied on data from the TC-STAR project and from the NIST MT evaluation. The TC-STAR corpus consists of proceedings of the European Parliament. It is a spoken language translation corpus containing the verbatim transcriptions of the speeches in the parliamentary sessions. The domain is basically unrestricted because a wide range of different topics is covered in the sessions. The translation direction is from Spanish into English. For the corpus statistics, see table A.4 in appendix A. The NIST task deals with translation of Chinese new articles into English. Yearly evaluation rounds are carried out by the National Institute of Standards and Technology (NIST). The goal is to set up a framework for comparing different (research) MT system against each other. The training data come from different news agencies and magazines, from the United Nations website, and from the proceedings of the Hong Kong Legislative Council. The corpus statistics are summarized in table A.5. The SMT systems whose output has been used for confidence estimation have been trained on these corpora. The same holds for the probability models that were used to estimate the word confidences.

Table 7.1: Translation quality of the alignment templates system (AT) and the phrase-based translation system (PBT) on the TT2 Xerox test corpora. Translation directions are Spanish → English, French → English, and German → English (see tables A.1 through A.3 for corpus statistics).

		WER[%]	PER[%]	BLEU[%]	NIST
S → E	AT	29.6	20.1	63.4	8.80
	PBT	26.1	17.5	66.9	8.98
F → E	AT	54.8	43.7	31.5	6.64
	PBT	54.9	43.4	31.3	6.62
G → E	AT	62.7	49.8	26.6	5.92
	PBT	61.6	49.6	25.7	5.72

Table 7.2: Translation quality of additional MT systems on the TT2 Xerox test corpora.

		WER[%]	PER[%]	BLEU[%]	NIST
S → E	Systran	78.0	62.3	23.4	4.77
F → E	Systran	81.5	71.7	12.5	4.23
G → E	Systran	79.2	66.4	12.0	4.09
	FST	63.2	50.4	26.5	5.79

Several (S)MT systems have been used for testing the confidence measures. A detailed analysis will be given for two of them: the phrase-based translation system described in section 3.3.3.1 (denoted as PBT in the tables), and the so-called alignment template system described in appendix C.1 (denoted as AT). They are both state-of-the-art SMT systems. Using these systems, single best translations, word graphs and N-best lists have been generated. The translation quality of both systems on the TT2 Xerox data in terms of different translation evaluation measures is given in table 7.1. One can see that the best results are obtained on Spanish to English translation, followed by French to English and German to English.

Two more translation systems have been used for comparative experiments: One is a statistical MT system which is based on a finite state architecture (FST). An overview of this system is given in appendix C.2, for a detailed description see [Kanthak et al. 05]. Additionally, translations generated by Systran [sys 05] have been used. Table 7.2 presents the translation error rates and scores for these two systems on the TT2 Xerox test corpora. Their hypotheses have been used to investigate whether the confidence measures proposed in this thesis perform well independently of the translation system which generated the translations.

All three SMT systems (AT, PBT and FST) show very similar performance on the TT2 Xerox data. The fact that Systran generates translations of much lower quality is due to the very specific terminology of the technical manuals. The SMT systems have been trained on similar corpora, so that they have learned the vocabulary and style of this domain.

Additional classification experiments have been performed on the EPPS corpora

Table 7.3: Translation quality of the phrase-based translation system on the EPPS Spanish → English test corpus, version: verbatim.

	WER[%]	PER[%]	BLEU[%]	NIST
without case	40.9	30.4	45.5	9.83
with case	42.5	32.2	45.1	9.67

Table 7.4: Translation quality of the phrase-based translation system on the NIST 2004 Chinese → English test corpus.

WER[%]	PER[%]	BLEU[%]	NIST
61.8	42.9	31.1	8.47

compiled in TC-STAR and on the Chinese-English corpora from the NIST MT evaluation campaign. The translations of development and test corpus have been generated by the phrase-based translation system in both cases. The translation quality is given in tables 7.3 and 7.4.

Scaling factors

For the system-based confidence measures which are calculated over word graphs or N-best lists, a global scaling factor is introduced. The sentence probability is weighted by this factor in the summation (see equation 3.6). The reason why this global factor is necessary is that the sub-model weights determined for the translation process express only the relation between the sub-models, and not the absolute values. The global scaling factor is optimized on the development corpus w.r.t. classification performance measured by one of the evaluation criteria presented in section 4.3. In the experiments presented in the following, it has been optimized with respect to CER.

For the direct phrase-based confidence measures, it is possible to optimize not only the global scaling factor, but to determine also the relation between the different sub-models anew. The sub-model weights which are optimal for translation are not necessarily the optimum for classification, so one can expect an improvement from this step. Again, the scaling factors have been optimized w.r.t. CER.

In principle, it is possible to optimize the sub-model weights for the system-based confidence measures as follows:

1. generate an N-best list and calculate the confidence measures,
2. perform classification and evaluation,
3. modify sub-model weights,
4. rescore the N-best list using the new weights and calculate the confidence measures.

Steps 2 to 4 are repeated until an optimum on the development set is found. Since the rescoring of the N-best list is computationally very expensive, this approach has not been pursued for this thesis.

7.2 Classification results

This section gives an overview of the experiments using word posterior probabilities as confidence measures in the setting described in chapter 4. The different word posterior probabilities presented in chapter 3 are studied in detail and compared to each other. The effect of the definition of word error on confidence estimation will be analyzed. The performance of the confidence measures is measured in terms of CER and IROC. For the baseline CER, the 90%- and 99%-confidence intervals are given. These have been determined with the bootstrap estimation method described in [Bisani and Ney 04][a].

Section 7.2.1 will present a detailed analysis of the system-based confidence measures. The different concepts of word posterior probabilities will be compared and evaluated w.r.t. reference tags defined by different word error measures. The confidence measures based on direct models will be investigated in section 7.2.2. An overview of the best methods on all considered language pairs will be presented in section 7.2.3. The influence of the scaling factor(s) on performance of the confidence measures will be studied in section 7.2.4. An excursion feature combination, including a description of the experiments conducted in the CLSP summer workshop 2003, will be given in section 7.2.5. Section 7.2.6 will report on comparative experiments between the direct phrase-based confidence measures and the system-based confidence measures when applied to output from the PBT system. Alternative definitions of the classifier based on confidence measures will be investigated in section 7.2.7. Section 7.2.8 will conclude the classification experiments.

7.2.1 System-based confidence measures

In this subsection, a detailed analysis of the different system-based confidence measures introduced in section 3.2 will be given. Their classification performance for all word error measures presented in section 4.2 will be analyzed. This will show how the different concepts of word posterior probabilities are related to the definitions of word errors. Additionally, the influence of the length of the N-best list on confidence estimation performance will be investigated.

Table 7.5 shows the classification performance in terms of CER for the different system-based confidence measures. Results are presented for all different word error measures. The definition of word error strongly affects CER: the baseline CER ranges between 27.5% and 42.2%. The results show that the word graph based word posterior probabilities outperform the ones calculated over N-best lists in all but one cases. However, the differences are not significant for WER, PER, and 'Set'. There is one exception where

[a] The tool is freely available from
`http://www-i6.informatik.rwth-aachen.de/web/Software/index.html`.

Table 7.5: Classification performance in terms of CER[%] for different system-based confidence measures on the TT2 Xerox French → English test set, version: raw. References based on the different word error measures introduced in section 4.2. Hypotheses from the phrase-based translation system. Note that all scaling factors have been optimized w.r.t. WER based classification. The best result for each word error measure is printed in boldface.

model	WER	PER	Set	Pos	2-gram	3-gram	4-gram
baseline	42.2	34.2	32.3	30.8	40.6	32.0	27.5
99%-confidence interval	±2.3	±2.0	±1.9	±3.0	±2.5	±2.7	±2.8
90%-confidence interval	±1.5	±1.2	±1.2	±1.9	±1.6	±1.7	±1.7
N-best lists, fixed position	39.7	33.4	31.5	22.5	30.9	24.5	20.8
Levenshtein	31.3	28.1	26.7	33.4	34.9	33.3	29.6
window ±3	31.6	28.3	26.7	32.7	35.5	33.1	31.9
average position	38.5	32.1	30.2	28.0	38.0	30.5	25.8
any position	38.0	31.3	29.4	28.2	38.5	30.8	26.1
count-based	31.9	**26.8**	26.7	34.1	35.2	33.2	35.6
context ±1	33.4	30.3	28.8	30.8	38.8	37.0	27.5
predecessor	36.5	32.7	32.3	30.8	37.2	34.4	27.5
word graphs, fixed position	39.1	33.0	31.5	**19.3**	**28.6**	**21.5**	**17.9**
window ±3	**31.1**	27.5	**26.0**	30.6	33.7	31.6	29.4

the confidence measure determined over N-best lists is better than all word graph based methods: For PER based classification, the best confidence measure is the one derived from the posterior risk as shown in section 5.3.3. For this confidence measure, no efficient way of calculating it over a word graph is known. Over N-best lists, however, the calculation can be easily performed. The best classification performance for reference classes defined by WER and 'Set' is achieved by summing over a window of target positions on word graphs. For the stricter word error measures – namely the ones which consider the fixed target position of the word or its predecessor(s) – the word graph based confidence measure without windowing performs best.

Comparing the different concepts of word posterior probabilities, it shows clearly how the performance depends on the underlying concept: for WER based classification, either the Levenshtein-based approach or the one using windowing (over word graphs or N-best lists) are the best measures. These are the methods which consider the target position of the word, but allow for some variation. If the target position is kept fixed in evaluation (Pos), this should naturally be done in confidence estimation as well: The related word posterior probability (computed over word graphs or N-best lists) outperforms the other variants clearly. The same holds for PER based classification: the count-based word posterior probabilities derived from the posterior risk for PER are clearly superior to the other confidence measures.

It is interesting to see that the methods which completely ignore the position (denoted by "average position" and "any position" in the table) show weak discriminative power,

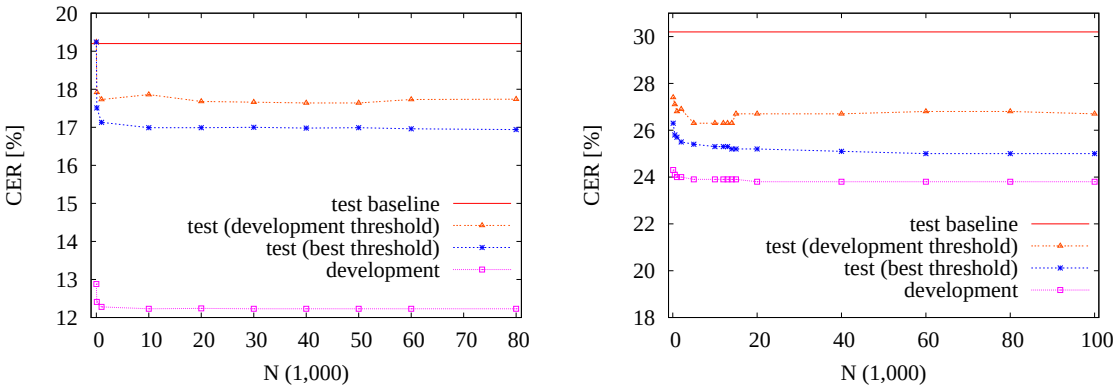


Figure 7.1: Effect of the N-best list length on CER[%] for the confidence measure based on Levenshtein alignment. TT2 Xerox Spanish → English, version: simplified (left), EEPS Spanish → English, version: verbatim (right). References based on WER. Hypotheses from the phrase-based translation system.

even if the reference tags are based on 'Set' or PER. Similarly, it seems that the predecessor- or context-based confidence measures do not capture the information relevant for predicting the n-gram error. A possible reason for this is that very rare words result in a high confidence for their successors because those will be the only words occurring in this context. Since these confidence measures perform badly, they have not been further investigated for this thesis.

The improvements achieved by the best confidence measures in table 7.5 are all significant at the 1%-level. They range from 20% to 37% relative. The improvement is the larger the stricter the word error measure is: The highest relative reduction of CER is achieved for the word error measure 'Pos' which considers only those words as correct which occur in the same position as they do in the reference. The lowest reduction of CER is achieved for the word error measure 'Set' which labels the highest number of words as correct as table 4.1 on page 47 shows.

All results presented in table 7.5 have been computed over N-best lists of length 10,000. Previous experiments showed that this length is sufficient to estimate the word posterior probabilities reliably. The effect of the length of the N-best list on the classification performance of the confidence measure is shown in figure 7.1 for the TT2 and EPPS Spanish–English corpora. The reference tags are given by WER, and the confidence measure is based on Levenshtein alignment. The minimal CER on the development and the test set as well as the actual CER achieved on the test set (with the threshold optimized on the development set) are plotted. On the TransType2 data, the CER drops significantly for increasing N-best list lengths up to 1,000. However, once the length exceeds 10,000, there is no noteworthy change in CER anymore. On the EPPS data, the CER decreases slightly if the threshold is optimized on the test set. However, if the threshold estimated on the development set is applied, the minimal CER is obtained for an N-best list length around 10,000.

7.2.2 Direct models

This subsection contains a detailed investigation of the different confidence measures based on direct models. First, the methods based on IBM model 1 (described in section 3.3.1) and on HMMs (see section 3.3.2) will be compared. Then, the direct approach to confidence estimation using phrases (introduced in section 3.3.3) will be studied in detail.

IBM model 1 and HMM

A detailed analysis of the HMM and IBM-1 based confidence measures has been performed on different TT2 Xerox tasks using translation hypotheses generated by the alignment templates system. The results are given in table 7.6. They show that in three out of four cases, the IBM model 1 outperforms the HMM. This can be explained by the fact that the IBM-1 based method does not restrict the word alignment at all. The HMMs, on the other hand, strongly favor one-to-one alignments. Their classification performance suffers if the considered language pair requires much reordering or word alignments which are not one-to-one. When comparing the different HMM variants, one can see that the monotone HMM over a reordered target sentence clearly outperforms the inverted model. The only exception is the Spanish-English language pair, but none of these improvements in CER is statistically significant. A manual analysis has shown that especially cases where a single Spanish word aligns to many words in English or vice versa cause problems. Among these are prepositions and modal verbs. On French → English and German → English, the HMMs reduce CER significantly over the baseline, even at the 1%-level. When translating from French, the monotone HMM even outperforms the method based on IBM model 1. This is foreseeable considering the fact that the HMM alignment model is well-suited for the language pair: most of the correspondences between French and English words are one-to-one. For a great part, the sentences are translated in monotone order, with exception of e.g. adjective-noun reordering. French-English is also the only language pair on which there exists a significant difference between a bigram and a trigram language model combined with the HMM: The CER can be reduced by 1.4% absolute for WER based reference tags.

On Spanish → English, the combination of the monotone HMM with a trigram language model has not been investigated because all HMMs performed badly. On German → English, there are no results available either, due to runtime and memory problems. For more details on this issue, see [Schoenemann 05].

The classification performance of the IBM model 1 is better for PER based than for WER based error tagging. This was to be expected because this measure does not consider the target position of a word at all. There is no clear predominance of one of the two IBM-1 variants. Their performance is very similar on all language pairs. The improvements over the baseline are significant at the 1%-level on French → English and German → English. On the Spanish-English language pair, the CER for the IBM model 1 lies at the lower bound of the 90%-confidence interval.

Table 7.6: Classification performance in terms of CER[%] for the HMM and IBM-1 based confidence measures on different TT2 Xerox tasks. References based on WER and PER. Hypotheses from the alignment templates system.

reference	model	E → S	S → E	F → E	G → E
WER	baseline	17.8	20.8	42.2	49.2
	99%-confidence interval	±1.9	±1.9	±2.3	±2.3
	90%-confidence interval	±1.2	±1.2	±1.5	±1.5
	monotone HMM & bigram	17.8	20.8	32.7	34.0
	trigram	17.8	-	**31.3**	-
	inverted HMM & bigram	17.5	20.3	36.1	37.4
	trigram	17.6	20.3	36.2	37.1
	IBM-1 (avg.)	**16.7**	**19.9**	34.1	33.4
	IBM-1 (max.)	**16.7**	20.0	34.1	**32.8**
PER	baseline	13.6	16.3	34.7	40.9
	99%-confidence interval	±1.5	±1.7	±2.0	±1.9
	90%-confidence interval	±0.9	±1.1	±1.3	±1.2
	monotone HMM & bigram	13.6	16.3	27.7	32.2
	trigram	13.6	-	**27.3**	-
	inverted HMM & bigram	13.3	16.1	31.2	33.8
	trigram	13.4	15.8	31.2	34.0
	IBM-1 (avg.)	12.6	**15.4**	28.2	**28.9**
	IBM-1 (max.)	**12.5**	15.5	29.2	30.5

Direct phrase-based confidence measures

Tables 7.7 and 7.8 present a detailed analysis of the direct phrase-based confidence measures on the TT2 data. They show that these methods achieve a significant reduction of CER even on the Spanish → English translation task. The baseline CER is relatively low on this task, and the IBM-1 and HMM based measures did not yield any significant improvement.

Table 7.7 contains an evaluation of the direct phrase-based confidence measures for reference tags defined by WER and PER. The contribution of the different sub-models in this approach is analyzed. To this end, the full model given in equation 3.25 is compared to a confidence measure without language model and one without word and phrase penalty. For each such confidence measure, the sub-model scaling factors have been optimized w.r.t. WER and PER based classification separately. The results show that in three out of four cases, omitting the language model does not harm performance (the only exception being WER based classification on French → English). This confirms the supposition that the main impact comes from the phrase models. Similarly, the confidence measures without word and phrase penalty classify equally well or slightly worse than the full model. The impact of the different sub-models will be discussed in further detail in section 7.2.4.

As mentioned in section 3.3.3.2, there exist two different possible ways of normalizing the direct phrase-based models. Results for both of them are presented in table 7.7:

Table 7.7: Classification performance in terms of CER[%] for the direct phrase-based confidence measures on the TT2 Xerox French → English and Spanish → English test sets. References based on WER and PER. Hypotheses from the phrase-based translation system. The scaling factors have been optimized w.r.t. WER and PER individually.

	F → E		S → E	
model	WER	PER	WER	PER
baseline	42.2	34.2	19.2	14.9
99%-confidence interval	±2.3	±2.0	±2.0	±1.7
normalization 1, all models	**30.6**	**27.4**	**16.5**	13.3
w/o LM	31.6	27.5	**16.5**	13.2
w/o penalties	31.2	27.7	16.9	13.2
normalization 2, all models	31.3	27.7	**16.5**	**13.0**

1. Normalize so that a word posterior probability $p(e \mid f_1^J)$ is obtained, with $\sum_{e'} p(e' \mid f_1^J) = 1$ (see equation 3.25).
2. Normalize by the sum of all phrase pairs matching the source sentence. This yields a probability distribution over the events *"e is contained in a translation of* f_1^J*"* versus *"e is* not *contained in a translation of* f_1^J*"*.

The last row in table 7.7 contains results for the second normalization strategy. The discriminative power of the confidence measure does not differ significantly from that of the first method. It performs slightly worse on French → English and slightly better on Spanish → English. In the following, the first normalization strategy will be applied, if not explicitly stated.

Table 7.8 shows how the direct phrase-based models perform for reference tags given by the other word error measures. Note that these models are optimized for WER based classification. The tendencies of the results are the same on both language pairs investigated here. They can be summarized as follows:

- The full model discriminates better than the confidence measures which exclude one of the sub-models. The difference is very small in some cases, but the full model never performs worse than one of the reduced models.
- For all word error measures except for 'Set', the second normalization strategy yields lower CER. The gain is up to 2.9% absolute. Obviously, the improvement is the larger the stricter the word error measure is.
- The direct phrase-based confidence measures are well-suited for classification w.r.t. WER and PER. For the stricter word error measures, namely Pos and the n-gram based error counting, they do not discriminate as well as the system-based word posterior probabilities (see table 7.5). This is foreseeable because the direct confidence measures completely disregard the position of the word. However, the CER is reduced significantly in all cases.

Table 7.8: Classification performance in terms of CER[%] for the direct phrase-based confidence measures on the TT2 Xerox French → English test set, version: raw. References based on the different word error measures introduced in section 4.2. Hypotheses from the phrase-based translation system. Note that all scaling factors have been optimized w.r.t. WER based classification. The best result for each word error measure is printed in boldface.

task	model	Set	Pos	2-gram	3-gram	4-gram
F → E	baseline	32.3	30.8	40.6	32.0	27.5
	99%-confidence interval	±1.9	±3.0	±2.5	±2.7	±2.8
	normalization 1, all models	**25.8**	28.1	32.9	28.7	26.7
	w/o LM	26.4	30.2	33.8	31.5	29.1
	w/o penalties	26.1	29.8	33.7	30.1	28.3
	normalization 2, all models	26.4	**26.3**	**32.8**	**27.2**	**24.7**
S → E	baseline	13.3	39.1	31.3	37.4	41.1
	99%-confidence interval	±1.5	±3.9	±2.9	±3.1	±3.0
	normalization 1, all models	**11.4**	31.3	26.3	28.1	30.5
	w/o LM	11.7	31.2	26.5	29.3	31.5
	w/o penalties	11.7	31.2	26.3	29.9	32.3
	normalization 2, all models	11.7	**28.5**	**26.2**	**27.3**	**27.6**

All results which will be presented in the following sections have been generated by the following phrase-based confidence measure: using all sub-models and normalization strategy 1 (see page 79).

7.2.3 Overview of different confidence measures

In the previous subsections, the different variants of the system-based and the direct model based confidence measures have been analyzed in detail. In the following, an overview of the classification results will be given which compares the best variants of all confidence measures on several different corpora. The hypotheses used for this comparative study have been generated by three of the RWTH statistical machine translation systems: the alignment templates system (see appendix C.1), the phrase-based translation system (see section 3.3.3.1), and the finite state transducer system described in section C.2. Additionally, hypotheses generated by Systran [sys 05] will be used as input.

TransType2 Xerox data

Table 7.9 compares the classification performance of several confidence measures on the TT2 French-English task. The CER and the IROC values are given for WER and PER based classification. Note that higher IROC values express better performance. It is interesting to see that in most of the cases, the tendencies are consistent for the two evaluation metrics: lower CER is accompanied by higher IROC.

Table 7.9: Overview of classification performance in terms of CER[%] and IROC[%] for different confidence measures on the TT2 Xerox French → English test set, version: raw. References based on WER and PER, confidence measures optimized accordingly. Hypotheses from the phrase-based translation system. The best result is printed in boldface.

Model	WER		PER	
	CER[%]	IROC[%]	CER[%]	IROC[%]
baseline	42.2	-	34.2	-
99%-confidence interval	±2.3	-	±2.0	-
N-best lists, fixed position	39.7	66.2	33.3	66.2
Levenshtein	31.3	72.6	28.1	74.8
count-based	31.9	71.6	**27.0**	**76.5**
word graphs, window ±3, sum	31.1	72.4	27.3	75.4
IBM-1 (max.)	39.2	67.0	31.5	71.0
monotone HMM & trigram	33.2	70.8	28.7	73.8
direct phrase-based	**30.6**	**74.4**	27.4	73.7

In general, one can see that the very simple approach which sums over sentences in the N-best list considering the fixed target position of the word clearly performs worst. This is to be expected, and the method has been included only for comparison. It can be considered as a simple baseline method.

The IBM-1 and HMM based confidence measures perform rather poorly compared to the other methods. This was to be expected since the IBM model 1 is a very simple model, and the HMMs suffer from the restrictions in the alignment model. The monotone HMM is superior to the IBM model 1 on the French → English task. This is consistent with the results reported in section 7.2.2 for translations generated by the alignment templates system.

The system-based confidence measures show much better discriminative power than the IBM model 1 and the HMM. The N-best list based measure with Levenshtein alignment and the word posterior probabilities calculated over word graphs perform similarly well. For WER based classification, they are outperformed only by the direct phrase-based approach which achieves the best CER and IROC values. The count-based method calculated over N-best lists is clearly the best confidence measure for PER based classification. Even if its CER does not differ much from that of the direct phrase-based measure, there exists a clear predominance in terms of IROC. Obviously, the correct classification threshold can be estimated very reliably for the direct phrase-based method, yielding low CER. But generally speaking, the count-based method seems to be superior. This result was to be expected because the count-based word posterior probability has been derived from the posterior risk for the PER.

The comparison of the IROC values for WER and PER based classification shows that PER is easier to learn than WER: The IROC values for PER are higher for most confidence measures. This is consistent with the results obtained in the

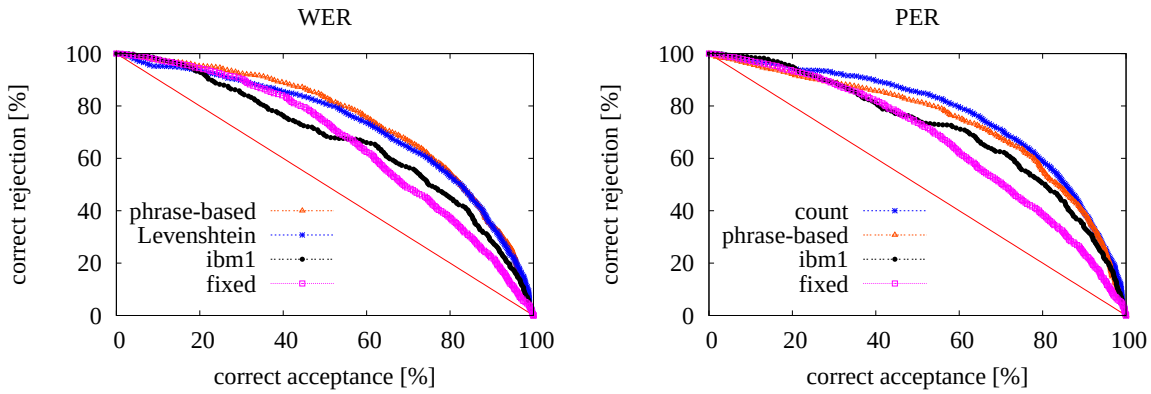


Figure 7.2: ROC curves for different confidence measures on the TT2 Xerox French → English test set, version: raw. References based on WER (left) and PER (right). Hypotheses from the phrase-based translation system.

CLSP workshop [Blatz et al. 03]: The classifiers investigated there also show better discriminative power for reference classes based on PER than for WER. It was found in the workshop that word error metrics which assign the same class (either correct or incorrect) to many words in the corpus are easier to learn. The same holds here: The baseline CER for WER is much closer to 50% than that for PER. This makes it harder for confidence measures to classify the words correctly w.r.t. WER.

To further illustrate the classification performance of the different confidence measures, the ROC curves for some of them are given in figure 7.2. The diagonal line refers to random classification of words as correct and incorrect. The left curve shows the results for WER based classification, and the right one for PER, respectively. The N-best list based method w.r.t. fixed target position is again given for comparison. One can see that the IBM-1 based confidence measure is clearly better than this baseline for PER, but not for WER. The curves for the direct phrase-based model and the best N-best list based method lie relatively close to each other. These two classifiers clearly dominate all others.

TransType2 Xerox: Results on translations generated by other MT systems

As stated in section 7.1, hypotheses from two additional machine translation systems have been used as input for the confidence estimation. Those hypotheses have been generated by the finite state transducer system (see section C.2), and by the Systran version available on the Altavista web-page in June 2005 [sys 05]. These experiments were performed to see whether the direct phrase-based confidence measures derived from the PBT system discriminate well on output from other systems. Especially the translations from Systran will yield insight on how much the classification performance depends on the underlying machine translation system. In this subsection, the results obtained on Systran translations on all three TT2 Xerox corpora will be presented. For the complete set of experiments on translations from all four available MT systems, see appendix D.

The evaluation of the system-independent confidence measures on the Systran output is

Table 7.10: Classification performance in terms of CER[%] and IROC[%] for confidence measures based on direct models. TT2 Xerox test sets. Reference based on WER. Hypotheses from Systran.

model	F → E		S → E		G → E	
	CER	IROC	CER	IROC	CER	IROC
baseline	32.8	-	43.7	-	37.4	-
99%-confidence interval	±1.7	-	±1.5	-	±1.4	-
IBM-1 (max)	26.0	81.3	21.7	85.5	**23.6**	80.7
HMM* & bigram	31.9	55.9	32.2	81.8	23.8	**81.5**
direct phrase-based	**22.7**	**83.2**	**17.3**	**87.5**	24.3	81.4

* : monotone HMM for French → English and German → English, inverted HMM for Spanish → English

shown in table 7.10. The investigated methods are the ones which consider only the single best translation and no additional system output: the confidence measures based on IBM model 1, HMMs and the direct phrase-based method. All measures distinctly decrease the CER compared to the baseline, with significance at the 1%-level. On Spanish → English and French → English, the three confidence measures can be clearly ranked: the HMM based confidence measures perform worst, followed by those based on IBM model 1. The direct phrase-based approach significantly outperforms the two other methods. These tendencies are consistent for both CER and IROC. On Spanish → English, the inverted HMM instead of the monotone model has been applied, because it performs slightly better. On German → English, all three confidence measures discriminate similarly well. In terms of CER, the IBM model 1 is slightly better than the other two, whereas the HMM achieves the highest IROC value.

TC-STAR EPPS data

Experiments comparing the classification performance of the different confidence measures have been carried out on the EPPS data which is structurally different from the Xerox manuals. The EPPS collection consists of speeches given in the plenary sessions of the European Parliament, translated from Spanish into English. The EPPS task is more challenging than the Xerox manuals because the domain is almost unrestricted and the translation has to cope with effects of spontaneous speech. The goal of these experiments is to find out whether the confidence measures perform equally well on this challenging task as on the Xerox data. The development and test set of the EPPS data are provided with two references each. This makes it possible to compare the two ways of handling multiple references: As explained in section 4.2, the true class of a word can be determined either w.r.t. the pooled references or to the reference with minimal distance.

Table 7.11 presents the CER and IROC for different confidence measures on the EPPS task. The classification w.r.t. m-WER and m-PER as word error measures has been investigated. It can be seen that the word posterior probabilities derived from the posterior risk for these error measures perform best: the Levenshtein-based confidence measure discriminate best for m-WER and the count-based approach for m-PER. Especially in

Table 7.11: Overview of classification performance in terms of CER[%] and IROC[%] for different confidence measures on the TC-STAR EPPS Spanish → English test set, version: verbatim. Reference based on m-WER and m-PER, confidence measures optimized accordingly. Hypotheses from the phrase-based translation system, lower case.

	m-WER		m-PER	
model	CER	IROC	CER	IROC
baseline	30.2	-	23.5	-
99%-confidence interval	±1.2	-	±1.0	-
N-best lists, Levenshtein	**25.7**	**75.4**	21.6	74.2
window ±3	26.7	69.9	21.4	71.8
count-based	27.6	71.4	21.2	**78.3**
IBM-1 (max)	27.7	68.7	21.5	72.5
direct phrase-based, normalization 1 (see p. 79)	26.8	67.5	21.2	70.9
normalization 2 (see p. 79)	26.9	69.0	**21.1**	70.1

terms of IROC, they are clearly superior to all other confidence measures. The CER values for m-PER do not differ largely for all investigated confidence measures.

The results achieved by the direct phrase-based approach on this task are not as good as on the Xerox data. The reason for this is that the domain of the EPPS collection is almost unrestricted. The phrase models do not capture the data as well as they do in the Xerox domain, as the analysis presented in section 7.2.6 will show. Nevertheless, for m-PER based classification, the direct phrase-based measures achieve the same reduction CER over the baseline as the system-based method using count information. Since the direct phrase-based confidence measures completely disregard the target position of the word, they are better suited for PER based classification than for WER. The two different normalization strategies presented on page 79 do not differ significantly in terms of CER or IROC. However, the system-based confidence measures outperform both of the variants.

As to be expected, the IBM model 1 based confidence measure performs better for reference tags defined by m-PER than for m-WER. However, it is among the methods with the worst discriminative power in both cases. The HMM based confidence measures have not been investigated on this task due to extremely high runtime and memory requirements.

In general, the improvement over the CER baseline are not as high on these data as on the TransType2 Xerox corpora. The gain in CER is 15% relative for the best confidence measure. But since the test corpora are large – with 20k running words they are about twice as big as the ones from the Xerox data – all the achieved improvements are significant at the 1%-level. The IROC values are comparable to those achieved on Xerox data. The fact that the IROC is independent of the baseline error allows for the conclusion that the confidence measures are well-suited for this challenging translation task as well.

The ROC curves shown in figure 7.3 further illustrate the classification performance of the different measures. The left curve shows the results for m-WER based classification,

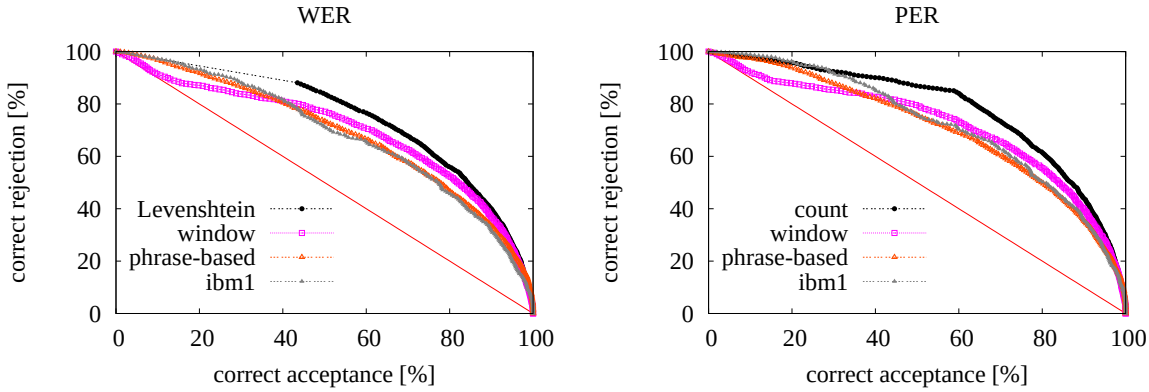


Figure 7.3: ROC curves for different confidence measures on the EPPS Spanish → English test set, version: verbatim. References based on m-WER (left) and m-PER (right). Hypotheses from the phrase-based translation system.

Table 7.12: Classification performance of different confidence measures for reference tags given by pooled WER and PER. CER[%] and IROC[%] on the TC-STAR EPPS Spanish → English test set, version: verbatim. Hypotheses from the phrase-based translation system, lower case.

	pooled WER		pooled PER	
model	CER	IROC	CER	IROC
baseline	23.5	-	18.5	-
99%-confidence interval	±1.1	-	±1.0	-
N-best lists, Levenshtein	21.3	**77.5**	17.0	76.4
window ±3	21.8	71.5	17.1	73.2
count-based	21.8	73.7	**16.7**	**80.6**
IBM-1 (max)	21.7	70.2	16.9	74.3
direct phrase-based, normalization 1 (see p. 79)	21.3	69.5	16.8	69.3
normalization 2 (see p. 79)	**21.2**	70.4	16.9	69.9

and the right one for m-PER, respectively. One can see that for m-WER, the IBM-1 based and the direct phrase-based confidence measures perform very similarly. There is no clear difference between these two approaches and the one performing windowing over target positions. The direct models discriminate better if CA is low, and the system-based measure for low CR, respectively. The Levenshtein-based word posterior probabilities are clearly superior to all other approaches. The ROC curve lies beyond the others over the whole range. For PER, the classifier based on word counts dominates all other confidence measures. The three other methods show a relatively similar performance.

For all results presented so far, the reference tags are determined by comparing the hypothesis to the most similar reference. As mentioned in section 4.2, it is also possible to pool the references instead. Table 7.12 presents an assessment of the discriminative power of different confidence measures for these reference tags. The conclusions from these

results are the same as for those in table 7.11: The Levenshtein-based method performs best for WER, and the count-based one for PER. All reported improvements in CER are significant at the 1%-level. The IROC values for the pooled error measures are higher than for m-WER and m-PER for all confidence measures. Obviously, this method of error counting is easier to assess using confidence measures. The differences in CER are not as big here as in table 7.11. However, the IROC values clearly distinguish the quality the classifiers.

NIST Chinese-English data

The third translation task which has been used for the evaluation of the confidence measures proposed in this thesis is part of the NIST MT evaluation campaign. The task here is the translation of news articles from Chinese into English. Similarly to the EPPS data, the domain is basically unrestricted. However, the vocabulary size and the training corpus are much larger than on the EPPS collection, as the corpus statistics presented in appendix A show.

The experimental results are presented in table 7.13 and figure 7.4. The confidence measures which performed best on the tasks investigated so far have been evaluated on the NIST data. The results are comparable to those achieved on the EPPS collection. All confidence measures reduce CER over the baseline with significance at the 1%-level. For reference tags defined by m-WER, the confidence measure using Levenshtein alignment over N-best lists performs best. Especially in terms of IROC, this method is clearly superior to the other confidence measures. The left plot in figure 7.4 shows that its ROC curve dominates all other curves. The count-based method achieves a CER that is 0.1% lower, which is not significant. For classification w.r.t. m-PER, there are two methods which outperform the others: the count-based confidence measure calculated over N-best lists and the direct phrase-based approach. They achieve CER and IROC values which differ significantly from those of the other measures. However, none of the two approaches is clearly superior to the other: The direct phrase-based confidence measure achieves lower CER of 27.4%, whereas the count-based confidence measure calculated over N-best lists achieves a slightly higher IROC value. The ROC curves of both methods in the right plot in figure 7.4 lie very close together. None of them dominates the other. The confidence measure based on IBM model 1 shows by far the worst discriminative power for both m-WER and m-PER based classification. The CER obtained with this method is significantly higher than those of all other measures. As the figure shows, the ROC curve for this confidence measure lies below those of all methods in both cases.

The results of the comparative experiments can be concluded as follows: The best performance is achieved be the confidence measures which are derived from the word error measure defining the reference class, and by the direct phrase-based method. This tendency is consistent across all different corpora and systems.

Table 7.13: Classification performance in terms of CER[%] and IROC[%] for different confidence measures on the NIST04 Chinese → English test set. Reference based on m-WER and m-PER, confidence measures optimized accordingly. Hypotheses from the phrase-based translation system.

	m-WER		m-PER	
model	CER	IROC	CER	IROC
baseline	46.2	-	32.7	-
99%-confidence interval	±1.0	-	±0.6	-
90%-confidence interval	±0.6	-	±0.4	-
N-best lists, Levenshtein	37.2	**67.4**	30.5	67.5
window ±3	39.2	64.4	30.5	67.0
count-based	**37.0**	66.0	28.4	**72.0**
IBM-1 (max)	42.9	58.0	31.9	59.9
direct phrase-based	37.3	66.7	**27.1**	71.8

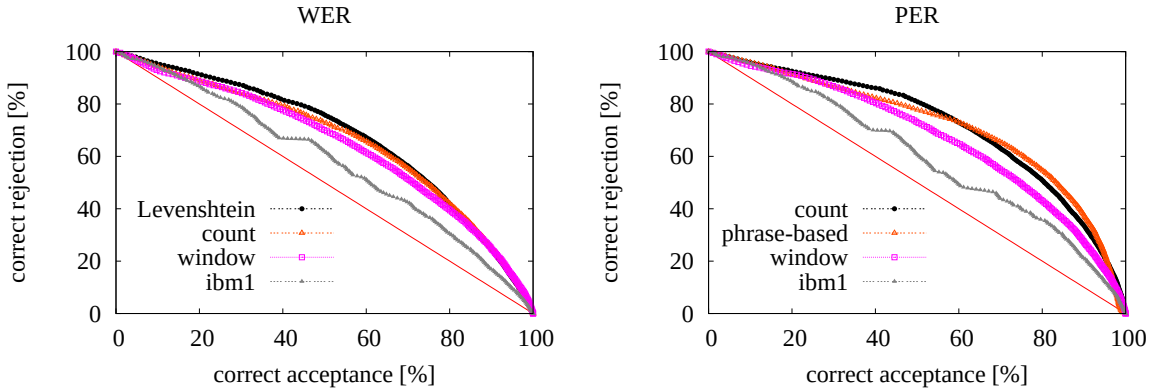


Figure 7.4: ROC curves for different confidence measures on the NIST Chinese → English test set. References based on m-WER (left) and m-PER (right). Hypotheses from the phrase-based translation system.

7.2.4 Scaling factors

For all experimental results presented in this thesis, the scaling factor(s) in the confidence measures have been optimized w.r.t. classification performance measured by CER. In this section, the impact of these scaling factors on confidence estimation will be investigated. First, the integration of a global scaling factor for the system-based confidence measures as shown in equation 3.6 on page 17 is studied. Table 7.14 shows the effect of a global scaling factor on classification performance on the TT2 Xerox French → English data. For each system-based confidence measure, the classification performance with and without additional global scaling factor is compared for the word error measures WER and PER. The scaling factor has been optimized on the development w.r.t. CER individually for each confidence measure, and then evaluation has been performed on the test set. The experiments have been performed for the most promising variants of the system-based

Table 7.14: Effect of the global scaling factor on classification performance in terms of CER[%] for different system-based confidence measures on the TT2 Xerox French → English test set, version: raw. References based on WER and PER. Global scaling factor optimized individually for each confidence measure. Hypotheses from the phrase-based translation system.

reference tag	WER		PER	
global scaling factor	1	opt.	1	opt.
baseline	42.2		34.2	
99%-confidence interval	±2.3		±2.0	
N-best lists, Levenshtein	33.9	31.3	29.3	28.1
window ±3	34.5	31.6	28.7	28.3
count-based	34.0	31.9	27.6	**27.0**
word graphs, window ±3	31.8	**31.1**	27.7	27.3

confidence measures: over N-best lists, the variants based on Levenshtein alignment, windowing over target positions, and the counts of the words have been calculated. Additionally, the best word graph based method has been investigated. This is the position-based word posterior probability, summed over a window of ±3 target positions.

For the confidence measures calculated over N-best lists, a significant gain is achieved through the optimization: the CER drops by up to 2.9% absolute. The improvement is larger for WER based classification than for PER. Without optimization, the N-best list based methods perform significantly worse than the word graph based confidence measure. But using the global scaling factor, their performance is very close to that of the word graph based word posterior probabilities. For the word posterior probabilities determined over word graphs, the CER is also reduced through the introduction of the global scaling factor. However, the gain is smaller here.

Figure 7.5 gives a detailed analysis of the effect of the global scaling factor on the classification performance of the confidence measures using Levenshtein alignment. The CER is plotted versus the global scaling factor on the development sets from the EPPS and the NIST tasks. The vertical lines mark the optimized factors. This value is 1.2 on the EPPS data and 7.1 on the NIST data, respectively. The CER can be reduced significantly through the optimization. Especially on the EPPS data, the curve has a distinct minimum. As one can see, the values of the global scaling factor are greater than 1 on both corpora. This indicates that the probabilities of the sentences in the N-best list are too close to each other to yield a reliable estimate of the word posterior probabilities. If the sentences contained in the list are very similar, their probabilities will not differ much in most cases. Scaling the sentence probabilities with a factor larger than 1 sharpens the distribution and assigns more probability mass to the top hypotheses. Obviously, the sentences in the lower ranks of the list do not contribute relevant information to the confidence estimation so that scaling down their probabilities improves confidence estimation.

For the direct phrase-based confidence measures, it is possible to optimize all sub-model

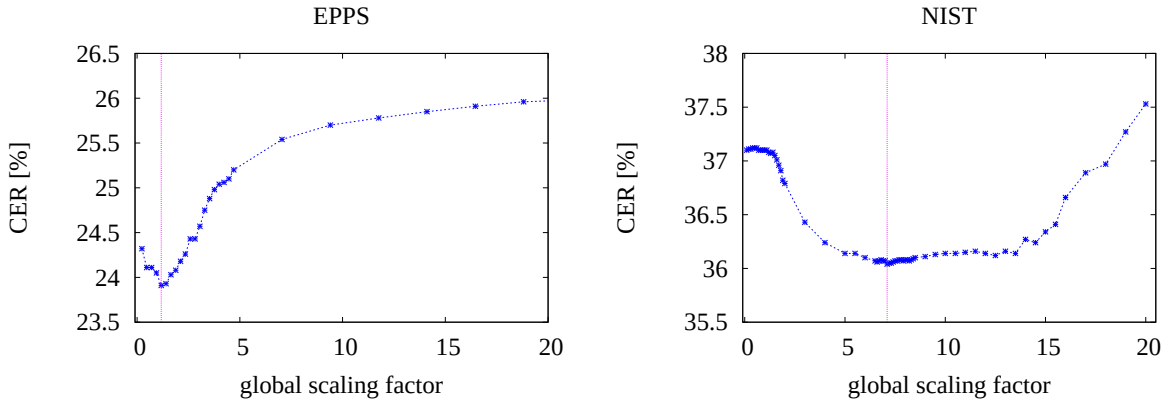


Figure 7.5: Effect of the global scaling factor on classification performance in terms of CER[%]. Results for the N-best list based confidence measure with Levenshtein alignment. EPPS Spanish → English (**left**) and NIST Chinese → English (**right**) development sets. References based on WER. Hypotheses from the phrase-based translation system.

Table 7.15: Effect of sub-model scaling factors on classification performance in terms of CER[%]. Results for the direct phrase-based confidence measure on the TT2 Xerox French → English test set, version: raw. References based on WER and PER. Hypotheses from the phrase-based translation system. The scaling factors have been optimized w.r.t. WER and PER individually.

reference tag		WER	PER
baseline		42.2	34.2
optimized for	translation	33.9	29.9
	confidence estimation	30.6	27.4

scaling factors (as explained in section 3.3.3.2). One experiment has been conducted which analyzes the contribution of this optimization to classification performance. The confidence measures have been calculated using the scaling factors which are optimal for translation, and using the values optimized w.r.t. CER. The results are shown in table 7.15. The effect of the optimization is even stronger in this case: For WER based classification, the CER drops by 3.3% absolute. What is interesting about this result is that the non-optimized confidence measure shows the same classification performance as the one based on N-best lists (see table 7.14). But with optimization, the direct method outperforms the system-based one. The effect is similar for PER based classification: If the direct phrase-based models are not optimized, they perform worse than the system-based confidence measures. But with optimization, they outperform all but one methods.

To illustrate the effect of the optimization, the actual values of the scaling factors are presented in table 7.16. For two different tasks and language pairs, namely the TT2 French → English corpus and the EPPS Spanish → English data, the values of the sub-model weights which have been optimized for translation and for confidence estimation

Table 7.16: Values of the sub-model scaling factors optimized for translation and for confidence estimation. Results for the TT2 Xerox French → English and EPPS Spanish → English data. References based on WER and PER. Hypotheses from the phrase-based translation system.

task	sub-model	optimized for	
		translation	confidence estimation
TT2 F → E	phrase lexicon $p(\tilde{e}\|\tilde{f})$	0.4750	1.0379
	phrase lexicon $p(\tilde{f}\|\tilde{e})$	0.0070	0.1899
	word lexicon $p(f\|e)$	0.0227	-0.0050
	word lexicon $p(e\|f)$	0.1116	0.0004
	word penalty	-0.1736	0.0074
	phrase penalty	0.3579	-0.2407
	language model	0.1993	0.0101
	global scaling factor	1.2	16.0
EPPS S → E	phrase lexicon $p(\tilde{e}\|\tilde{f})$	0.1915	0.4311
	phrase lexicon $p(\tilde{f}\|\tilde{e})$	0.0898	0.1192
	word lexicon $p(f\|e)$	0.3028	0.0689
	word lexicon $p(e\|f)$	0.3740	0.1156
	word penalty	-0.2970	-0.2463
	phrase penalty	0.1400	0.1238
	language model	0.1989	0.3878
	global scaling factor	1.2	3.0

are given. The values are normalized so that they sum up to 1. For both settings, a global scaling factor is given. In the left column, this is the global scaling factor which has been determined for the system-based confidence measures with Levenshtein alignment. The right column shows the value which has been optimized using the direct phrase-based confidence measure. It is interesting to see that on the TT2 Xerox data, the main impact comes from the phrase models: The two phrase translation models and the phrase penalty are assigned high weights, whereas all other sub-models have weights close to 0. The effect is less drastic on the EPPS data. But one can still see that the phrase models obtain much higher weights than the word models: The contribution of the two word lexica and the word penalty are significantly reduced in comparison with the setting optimal for translation.

In order to analyze the impact of the different sub-models in the direct phrase-based approach in more detail, another set of experiments has been performed. The full model, optimized for WER, has been taken as starting point. Each of the sub-models has been omitted once, and classification with this reduced model has been carried out. The scaling factors for the six remaining sub-models have been left untouched. This is different from the experiments reported in tables 7.7 and 7.8 in section 7.2.2, where the weights have been optimized again. Table 7.17 presents the classification error rates for the reduced models on the French-English language pair from the TT2 Xerox corpus and on the Spanish-English EPPS data. The models are sorted by CER on French-English. The

Table 7.17: Contribution of the different sub-models for the direct phrase-based confidence measures to classification performance in terms of CER[%]. Results on the TT2 Xerox French → English and EPPS Spanish → English test sets. References based on WER. Hypotheses from the phrase-based translation system.

		TT2 F → E	EPPS S → E
baseline CER		42.2	30.2
omitted sub-model	none	30.6	26.8
	phrase penalty	30.9	26.8
	word lexicon $p(f\|e)$	30.9	26.9
	phrase lexicon $p(\tilde{f}\|\tilde{e})$	31.1	27.1
	word penalty	31.6	26.8
	word lexicon $p(e\|f)$	31.7	27.0
	language model	32.3	27.3
	phrase lexicon $p(\tilde{e}\|\tilde{f})$	33.3	28.3

tendencies of the results are similar on both corpora. One can see that the phrase penalty has the lowest impact on classification performance: The CER hardly changes if this is omitted. The sub-model which contributes most to the confidence estimation is the phrase lexicon in the direction $p(\tilde{e}|\tilde{f})$: The CER increases significantly on both corpora if this sub-model is not considered. The second most important model is the language model. In general, one can see that the models assigning probabilities to source words/phrases given a target word/phrase have a low impact on classification performance. This does not come unexpected when estimating the posterior probability of a target word given the source sentence.

7.2.5 Combination of confidence features

In the following, results from the experiments performed during the CLSP summer workshop 2003 on confidence estimation for machine translation will be reported. In this workshop, the combination of different features into one confidence measure was studied. The combination was performed using a naive Bayes classifier [Sanchis 04] and multi-layer perceptrons (MLPs) [Collobert et al. 02]. The experiments are shortly outlined in section 4.4. For a detailed description see [Blatz et al. 03].

The experimental setting used in the workshop was slightly different from that of the other experiments presented here. The important differences are the following:

- The additional confidence estimation layer (the MLP or the naive Bayes classifier) is trained on a labeled training corpus and optimized on a development corpus. In contrast to this, the word posterior probabilities investigated in this thesis do not have to be trained and can directly be interpreted as probabilities of correctness.
- The confidence estimation is performed for all words in the N-best list. In all other

Table 7.18: Classification performance in terms of minimal CER[%] and IROC[%] for single features and their combination using naive Bayes. NIST 2002 Chinese → English test set. References based on PER. N-best lists from the ISI alignment templates system.

feature		CER[%]	IROC[%]
baseline		36.2	–
word posterior probability and frequencies	any	30.8–30.9	70.6–70.7
IBM-1	average	31.2	69.9
word posterior probability and frequencies	source	31.9	69.4–69.5
word posterior probability and frequencies	fixed	32.5–32.7	68.6–68.9
SMT model based		33.1	67.1–67.3
simple target based		33.1	66.6–66.8
semantic features		33.2–33.4	66.7–66.8
combination of top 3 features		29.2	73.3
combination of all features		29.6	73.6

experiments reported in this thesis, the classification performance of the confidence measures was evaluated on the single best hypotheses.

- The CER reported for the workshop experiments is the minimal value on the test set. In the other experiments presented in this thesis, the acceptance threshold is optimized on the development set, and then evaluated on the test set in order to obtain more objective assessment of the confidence measure.

- During the workshop, the word posterior probabilities were calculated over N-best lists without an additional global scaling factor. The sentence probabilities computed by the underlying SMT system were used without modification.

Using the naive Bayes classifier, the performance of each individual feature for word-level confidence estimation was studied. Additionally, the best 3 and all 17 features were combined with the same classifier. Table 7.18 shows the classification performance of single features in terms of CER and IROC. The word error measure PER was used for labeling words as correct or incorrect. The models are sorted by their classification performance in terms of CER and IROC. They are grouped as follows: "word posterior probability and frequencies" denotes the word posterior probabilities as defined in section 3.1, the relative frequency in the N-best list (see equation 3.4), and the rank weighted sum (see equation 3.5). These values were calculated based on different conditions: the occurrence in any target position ("any"), the aligned source position(s) ("source"), and the fixed target position ("fixed") of the word. Furthermore, the features based on the underlying SMT model which generated the translations are grouped into one entry in the table. The semantic features and the simple target based features are gathered into two groups. The features within each of these groups show very similar performance.

The features which yield the best results are the system-based confidence measures based on occurrence of the word in any position in the target sentence. They give a

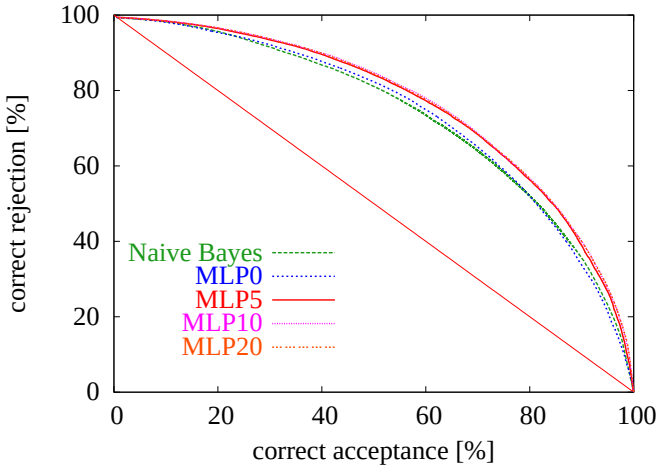


Figure 7.6: ROC curves for MLPs with different numbers of hidden units and the naive Bayes classifier. Combination of all 17 features. NIST 2002 Chinese → English test set. Reference tags based on PER.

significant improvement over the baseline of more than 5% absolute in CER as table 7.18 shows. The feature based on IBM model 1 discriminates similarly well. This is followed closely by the confidence measures calculated with regard to the aligned source position(s). The word posterior probabilities based on the fixed target position perform worse than the other word posterior probabilities, but still better than all other features except for IBM model 1. The SMT model based features as well as both types of target language based features perform rather poorly. They reduce CER by 3% absolute over the baseline. Three of the best performing features were combined using the naive Bayes classifier: the word posterior probabilities w.r.t. the aligned source positions and to occurrence in any target position, and the IBM-1 based feature. This combination yields a significant improvement over the performance of each single feature in terms of both CER and IROC. However, there is no significant change in CER or IROC if more information is added by combining all 17 features.

In addition to the naive Bayes classifier, several different MLP architectures were investigated for feature combination. The number of hidden units ranges from 0 to 20. Figure 7.6 compares the classification performance for different MLP architectures and for the naive Bayes classifier. Both include all 17 features. One can see that the naive Bayes classifier and the MLP with zero hidden units have a very similar performance. But as soon as the MLP gets more complex by the addition of more hidden units, the MLP outperforms the naive Bayes approach. There is no significant difference between the MLPs consisting of 5 to 20 hidden units; and their ROC curves can hardly be distinguished.

Since the feature combination yields good results in the experiments performed during the CLSP workshop, similar experiments have been performed for this thesis. The confidence measures investigated here have been combined log-linearly as described in section 4.4. The resulting confidence measures have been evaluated on all three translation tasks: the TT2 Xerox corpora, the EPPS and the NIST data. The three single word posterior probabilities which perform best in each setting have been used in

the combination. For the confidence estimation w.r.t. reference tags defined by m-WER, these are

- the system-based word posterior probabilities based on Levenshtein alignment,
- the system-based word posterior probabilities performing windowing over target positions,
- the direct phrase-based method.

If the reference tags are determined by m-PER, the features differ slightly between the different corpora. The measures which are combined are three out of the following:

- the system-based word posterior probabilities based on the word count,
- the system-based word posterior probabilities performing windowing over target positions,
- the confidence measure based on IBM model 1,
- the direct phrase-based method.

The experimental results for the combined confidence measures are presented in table 7.19. They show that the resulting confidence measure outperforms the best single method. The improvement is up to 1.6% absolute in CER. In terms of IROC, the gain is up to 4.1 points. This is in the same range as the improvements achieved in the CLSP summer workshop. However, there is one case in which the IROC decreases. This can be explained by the fact that the combination has been optimized w.r.t. CER. In order to avoid this type if inconsistency, the optimization could be performed considering a combination of CER and IROC as criterion.

7.2.6 System-based versus direct phrase-based confidence measures

When comparing the performance of the different confidence measures on the TT2 Xerox corpora, one can see that the direct phrase-based measures perform significantly better than those relying on N-best lists. This is the case even if the underlying N-best lists have been generated by the same phrase-based translation model. This fact is remarkable because the direct phrase-based confidence measures do not consider context at all and can thus be expected to yield a worse estimate of the word posterior probability. On the other hand, they allow for optimizing the scaling factors of the different sub-models w.r.t. classification performance. This optimization improves the classifier significantly as the results presented in section 7.2.4 show. In order to analyze which of the details of the two approaches is responsible for the difference in performance, a set of comparative experiments has been performed on the TT2 data. They are guided by the considerations presented in section 3.3.3.3:

Table 7.19: Classification performance in terms of CER[%] and IROC[%] for a log-linear combination of word posterior probabilities. TT2 Xerox French → English, EPPS Spanish → English, and NIST04 Chinese → English test sets. References based on m-WER and m-PER. N-best lists from the phrase-based translation system.

reference tag		m-WER		m-PER	
task	confidence measure	CER[%]	IROC[%]	CER[%]	IROC[%]
TransType2 F → E	baseline	42.2	–	34.2	–
	best single	30.6	74.4	27.0	76.5
	combination of 3	29.5	75.5	25.4	78.4
EPPS S → E	baseline	30.2	–	23.5	–
	best single	25.7	75.4	21.2	78.3
	combination of 3	25.7	76.1	20.1	76.9
NIST C → E	baseline	46.2	–	32.7	–
	best single	37.3	67.2	27.4	71.3
	combination of 3	35.5	68.0	25.8	75.7

1. An N-best list is generated using the sub-model scaling factors which have been optimized for confidence estimation. Then, the N-best list based confidence measures are calculated.

2. Rescoring of the translation N-best list with the sub-model weights optimal for confidence estimation is performed. Then, the N-best list based confidence measures are calculated.

3. Analogous to 2, but the context of the phrase is not considered. Each matching target phrase is taken into account exactly once, no matter how often it occurs in the N-best list. This is equivalent to filtering all matching phrases over the N-best list and using the direct phrase-based confidence measures with this restricted phrase lexicon.

The N-best list based confidence measure used in experiments 2 and 3 sums the probability of a sentence once for each occurrence of the word in this sentence. This is a slight variation of the confidence measure described in section 3.2.1.5 which takes each sentence probability into account at most once, no matter how often the word occurs in this sentence. Here, the probability of one sentence might be summed several times. This is done in order to mimic the behavior of the direct phrase-based model which sums the probability of each phrase containing the word.

All experiments reported in this subsection have been performed on the TT2 Xerox Spanish-English corpus. The translations have been generated by the phrase-based translation system, and the N-best lists have a maximal length of 10,000. The reference classes of the words are based on WER.

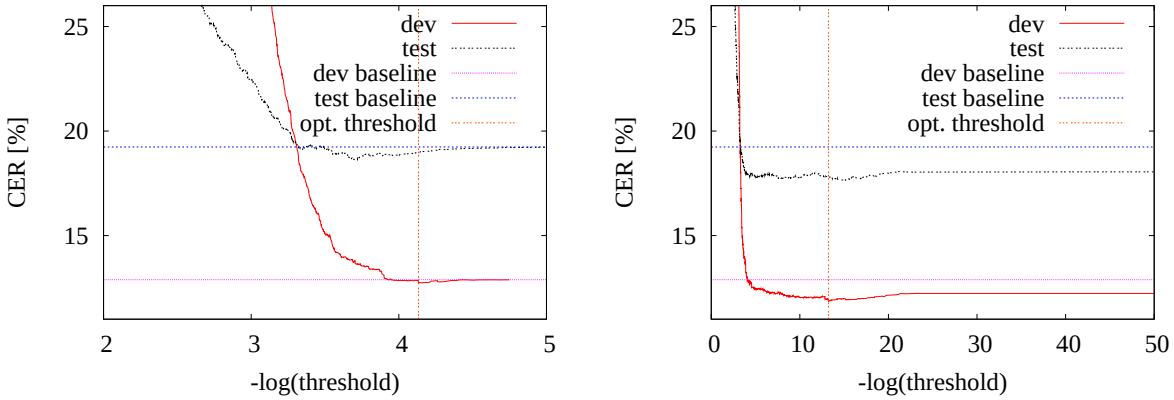


Figure 7.7: CER[%] as function of the threshold. TT2 Xerox Spanish → English development set. References based on WER. Hypotheses from the phrase-based translation system. N-best list based confidence measure any target position.
Left: N-best list generated with scaling factors optimized for WER.
Right: N-best list generated with scaling factors optimized for CER.

Ad 1: Generation of a new N-best list

Figure 7.7 compares the classification performance across different threshold values for the N-best list based confidence measure. The CER is plotted versus the negated logarithm of the threshold value on the development and test set. The left plot shows the results obtained with the original N-best list which has been generated to optimize translation quality. The right plot depicts the CER for the N-best list generated with the weights which are optimal for confidence estimation. These weights have been optimized w.r.t. CER using the direct phrase-based model. Then, the N-best list has been created with these optimized values. Note that the translation alternatives in this list will be different from those in the original list.

It can be seen in figure 7.7 that the confidence measure calculated over the original N-best list does not significantly reduce CER over the baseline. The values on the CER curve are above the baseline for most thresholds. Moreover, the minima for the two curves on the development and the test set do not lie in the same area. Thus, the threshold which is optimal for the development set is not well-suited for the test set. The picture looks completely different for the confidence measure determined over the N-best list which is generated with the weights optimal for confidence estimation: The CER can be significantly decreased in comparison with the baseline on both development and test set. The minima of the curves lie close to each other on both sets. The acceptance threshold estimated on the development set will thus be a good choice for the test set as well.

The CER on the test set for all comparative experiments described above is presented in Table 7.20. One can see that the gain from using the newly generated N-best list is 1.4% absolute in CER, which is significant at the 10%-level.

Ad 2: Rescoring the N-best list

In the second comparative experiment, the original N-best list has been rescored with the sub-model weights which are optimal for confidence estimation. For each sentence in the list, two separate scores have been determined: the language model probability together with the word penalty (as defined in equation 3.23 on page 39) and the phrase translation score (see equation 3.24). The product of these two scores is similar to the score assigned by the direct phrase-based confidence measure. Note that here, these scores are calculated over the whole sentence, and not only over a single phrase. The separation of the phrase and the language model score makes it possible to analyze the contribution of each model individually.

Table 7.20: Effect of the different model components on CER[%]. N-best list based vs. direct phrase-based confidence measure. TT2 Xerox Spanish → English test set. References based on WER. Hypotheses from the phrase-based translation system.

baseline			19.2
99%-confidence interval			±2.0
90%-confidence interval			±1.3
N-best list	generation optimized for WER		18.9
N-best list	generation optimized for CER		17.8
N-best list	rescoring optimized for CER	phrase + LM score	17.4
		only phrase score	17.3
		only LM score	18.3
phrase without context			16.6
direct phrase-based			16.5

Table 7.20 shows that rescoring the translation N-best list yields an improvement of 1.5% absolute over the experiment using the original N-best list. The CER reduction over the baseline is significant at the 1%-level for this method. Rescoring thus achieves better confidence estimation results than generating a new N-best list with the optimized scaling factors. There is no significant difference between the rescoring models with and without language model score. For comparison, also rescoring with the language model score only has been performed. As to be expected, this performs significantly worse.

Ad 3: Omitting the context

The next step bringing the two different approaches to confidence estimation closer together is to omit the sentence context. The direct phrase-based confidence measures consider only the score of one bilingual phrase at a time and completely disregard the rest of the sentence. To mimic this behavior, the phrase pairs matching the source sentence and the target sentence are determined for each translation hypothesis in the N-best list. The matching target phrases are collected for all translations of the same source sentence. To calculate the confidence of a target word e, the phrase and language model scores of all those target phrases containing e are summed. This is analogous to filtering

the set of all possible phrase pairs over the N-best list and then summing their scores as described in section 3.3.3.2. It can be seen in table 7.20 that this procedure yields another improvement in terms of CER. There is no significant difference to the CER achieved by the direct phrase-based confidence measure anymore.

The results of the comparative experiments can be concluded as follows:

- The optimization of the scaling factors w.r.t. classification performance plays the most important role. Rescoring of the N-best list with the optimized factors yields the largest gain in terms of CER.
- The phrase model has the main impact on confidence estimation as the rescoring experiments show. Omitting the language model does not harm the classification performance of the resulting confidence measure, but omitting the phrase model does.
- Omitting the context of the phrase improves the confidence estimation model. Nevertheless, this might be characteristic of the TT2 Xerox corpora which are very specific in terminology and sentence structure. On the EPPS and NIST data, the direct phrase-based confidence measures do not perform as well. Therefore, a data analysis has been performed which will be presented in appendix D.2.

7.2.7 Alternative definitions of the acceptance threshold

This subsection describes experiments on the different types of classifiers introduced in section 4. Apart from determining a global threshold – which is the standard approach – , thresholds based on the source or target sentence length have been investigated. A data analysis has been performed beforehand to investigate the distribution of the sentence lengths on several corpora. Only if these distributions are similar for development and test set, a gain can be expected from estimating separate thresholds for different sentence lengths. Appendix D.2.2 presents the results of the analysis. There are two language pairs where an improvement can be expected: French-English and German-English. In the following, a detailed analysis of the proposed methods will be given on French-English. Comparative results on three other data sets will be shown.

Separate thresholds for different sentence lengths

First, a separate threshold for each sentence length has been estimated as described in section 4.5. The source sentence length as well as the target hypothesis length have been used as indicator. Smoothing over the sentence lengths has been performed over a window of ± 4. The results from these experiments are presented in Table 7.21. Three structurally different confidence measures have been used: the direct confidence measures based on the phrase-based translation models and on IBM model 1, and the confidence measures calculated over N-best lists using Levenshtein alignment. The word errors have been counted using WER. In the first block, the classification performance in terms of CER is shown for a classifier using a global threshold for all words. The second block

Table 7.21: Effect of separate acceptance thresholds for different sentence lengths on classification performance. CER[%] for different confidence measures on the TT2 Xerox French → English development and test sets, version: raw. Reference based on WER. Hypotheses from the phrase-based translation system. Global acceptance threshold and separate thresholds for each source/target sentence length (smoothed a window of ±4).

threshold based on ...	confidence measure	DEV	TEST
baseline		34.9	42.2
99%-confidence interval		±2.1	±2.3
global threshold	direct phrase-based	26.7	30.6
	IBM-1 (max)	33.7	39.2
	N-best lists (Levenshtein)	27.8	31.3
... target hypothesis length	direct phrase-based	26.3	31.3
	IBM-1 (max)	32.4	35.2
	N-best lists (Levenshtein)	27.7	31.8
... source sentence length	direct phrase-based	26.3	**28.8**
	IBM-1 (max)	32.2	**33.0**
	N-best lists (Levenshtein)	27.3	**29.2**

shows the results for thresholds based on the translation hypothesis length. As to be expected, the performance improves on the development set, but it declines on the test corpus. Therefore, the idea of basing the acceptance on the hypothesis length has not been further pursued. But looking at the classifier based on source sentence length, the picture is totally different: The CER is significantly reduced on both development and test set. It drops by around 2% absolute on the test corpus for all three confidence measures. So the source sentence length seems to be a good indicator for classification.

Since the results for a classifier based on the source sentence length are encouraging on the French → English corpus, additional experiments have been performed on the other corpora. The setting which performed optimal on French → English has been chosen, i.e. the source sentence length based classification with smoothing over a window of ±4. Table 7.22 shows the results of these experiments.The same three confidence measures as before have been investigated. As to be expected from the data analysis, there is no (significant) improvement on the Spanish → English translation tasks. Only the CER for the IBM-1 based confidence measure drops by 0.6% on the EPPS data. In all other cases, the CER is lower if a global acceptance threshold is used. On TT2 Spanish → English, the CER even increases if length based thresholds are applied.

Threshold as a function of the sentence length

Another way of taking the length of source or target sentence into account is to modify the threshold by a function of this length as presented in equation 4.2 on page 53. Several different functions have been investigated: the (square) root, the identity function, the n-th power for $n \geq 2$, and the logarithm. However, none of these yielded any improvement

Table 7.22: Effect of separate acceptance thresholds for different sentence lengths on classification performance. CER[%] for different confidence measures on the following corpora: TT2 Xerox German → English (raw) and Spanish → English (simplified); EPPS Spanish → English (verbatim). References based on m-WER. Hypotheses from the phrase-based translation system. Global acceptance threshold and separate thresholds for each source sentence length (smoothed over a window of ±4).

corpus	threshold	confidence measure	DEV	TEST
TT2 G → E	baseline		38.3	48.4
	99%-confidence interval		±2.7	±2.4
	global threshold	direct phrase-based	26.4	**26.4**
		IBM-1 (max)	33.9	**32.8**
	separate thresholds	direct phrase-based	26.5	27.9
		IBM-1 (max)	33.9	36.1
TT2 S → E	baseline		12.9	19.2
	99%-confidence interval		±0.7	±2.1
	90%-confidence interval		±0.5	±1.3
	global threshold	direct phrase-based	11.0	**16.5**
		IBM-1 (max)	12.4	**18.3**
		N-best lists (Levenshtein)	12.2	**17.9**
	separate thresholds	direct phrase-based	10.8	20.1
		IBM-1 (max)	12.3	21.5
		N-best lists (Levenshtein)	12.2	21.0
EPPS S → E	baseline		27.1	30.2
	99%-confidence interval		±1.0	±1.2
	global threshold	direct phrase-based	25.6	**27.0**
		IBM-1 (max)	25.5	27.7
		N-best lists (Levenshtein)	24.2	**25.4**
	separate thresholds	direct phrase-based	25.3	27.3
		IBM-1 (max)	25.2	**27.1**
		N-best lists (Levenshtein)	23.9	26.0

in terms of CER. An example of the performance of the classifier for $h(J) = \frac{1}{\sqrt{J}}$ is given in table 7.23: the effect of the threshold modification on CER on the TT2 French → English task is shown. The investigated confidence measures are the same three used throughout this section. The table shows that the only confidence measure which can be improved through this modification of the threshold is the one based on IBM model 1. The CER drops by 4% absolute. However, the error rates achieved by the two other confidence measures are still significantly lower than this. The direct phrase-based approach as well as the one based on N-best lists are not improved through the modification. For the latter, the CER even increases by 4.2%. So generally speaking, this modification of the acceptance threshold does not yield a gain in classification performance.

Table 7.23: Effect of modifying the acceptance threshold on CER[%]. The threshold has been modified using a function of the source sentence length. Results for different confidence measures on the TT2 Xerox French → English development and test set, version: raw. References based on WER. Hypotheses from the phrase-based translation system.

classifier	confidence measure	DEV	TEST
baseline		34.9	42.2
99%-confidence interval		±2.1	±2.3
90%-confidence interval		±1.4	±1.5
global threshold (equation 4.1)	direct phrase-based	26.7	**30.6**
	IBM-1 (max)	33.7	39.2
	N-best lists (Levenshtein)	27.8	**31.3**
modified threshold (equation 4.2)	direct phrase-based	26.9	30.8
	IBM-1 (max)	33.2	**35.2**
	N-best lists (Levenshtein)	30.0	35.5

Acceptance of a fixed number of words

Instead of applying a global or sentence dependent threshold on the confidence of a target word, it is also possible to restrict the number of target words that should be accepted (see equation 4.3 on page 53). That is, all target words are sorted by their confidence and only the ones with the highest confidence values are accepted.

Two different ways of modeling the number of words to be accepted have been investigated in this thesis: The first approch accepts $I + w$ or $J + w$ target words, where I is the hypothesis length, J is the source sentence length, and w is an offset. The performance of this classifier is depicted in the left plot in figure 7.8. The minimal CER on the development is plotted versus the offset w for the direct phrase-based confidence measure. It can be seen that the performance of the modified classifier is always worse than that of the original method (denoted as "optimized threshold"). The difference is significant: The CER increases by 3.4% absolute for the classifier using the target sentence length, and 4.5% for the one based on the source sentence length. The second method accepts $h(J)$ target words, where J is the source sentence length. An example of such a function is $h(J) := s \cdot J$ for some $s > 0$. The right plot in figure 7.8 shows the results obtained with this type of classifier: The minimal CER on the development is plotted versus the scaling factor s. Again, it can be seen that the performance of the original classifier is by far better.

7.2.8 Conclusion

The results of the classification experiments can be summarized as follows:

- The optimization of the scaling factor(s) is crucial for classification performance of the confidence measures. This holds for the global scaling factor applied for the

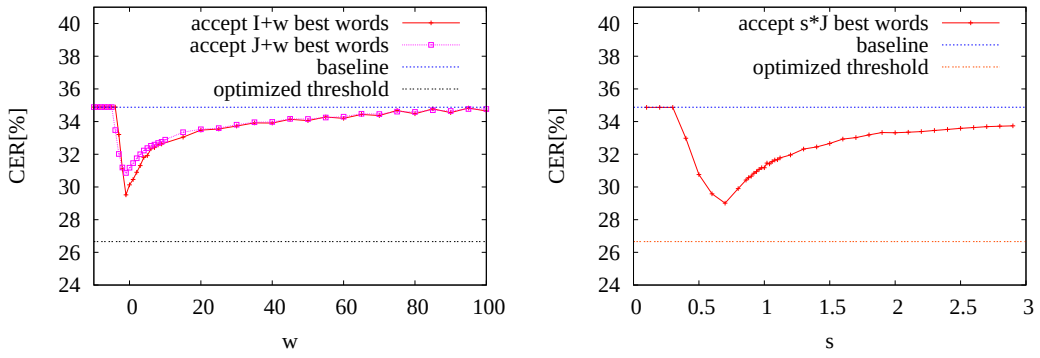


Figure 7.8: Effect of the acceptance of a fixed number of words on classification performance. **Left:** accept the $I+w/J+w$ words with the highest confidence, **right:** accept the $s \cdot J$ words with the highest confidence. Effect of w and s on minimal CER[%]. Results for the direct phrase-based confidence measure on the TT2 Xerox French $\rightarrow$ English development set. References based on WER. Hypotheses from the phrase-based translation system.

system-based confidence measures as well as for the sub-model weights of the direct phrase-based method.

- The performance of the confidence measures depends strongly on the word error measure which defines the reference tags. Naturally, the word posterior probabilities derived from the posterior risk for this word error measure discriminate best.
- In general, confidence estimation w.r.t. either very strict or very relaxed word error measures yields better results. The IROC values achieved by the best confidence measures are higher for PER than those for WER, for example.
- The combination of several different word posterior probabilities into one confidence measure yields better confidence estimation performance than the best single feature. This has been shown in experiments conducted during the CLSP summer workshop 2003 as well in experiments combining only confidence measures proposed in this thesis.
- The word posterior probabilities investigated in this thesis proved to be the best stand-alone features in the experiments conducted during the CLSP summer workshop 2003 [Blatz et al. 03].
- The confidence measures based on IBM model 1 and HMMs normally perform worse than the system-based or direct phrase-based methods. The reasons for this are the following: IBM model 1 is a very simple model which does not consider the context of a target word at all. The HMM based methods suffer from the restriction in the alignment model which favors one-to-one mappings of source and target words.
- The direct phrase-based confidence measures perform very well on restricted domains. On the TransType2 Xerox corpora consisting of technical manuals, they outperform all other measures. However, this is not the case for data from domains

which are basically unrestricted, such as the EPPS and NIST corpora. There, the system-based measures discriminate better for reference tags given by WER. For PER based confidence estimation, the direct phrase-based confidence measure and the count-based confidence measure calculated over N-best lists show the best performance.

- The direct phrase-based measures perform best if all sub-models are included. The sub-models which assign probabilities to target words/phrases given a source word/phrase have the highest impact on confidence estimation.
- Alternative ways of defining the classifier, based e.g. on source sentence length, improve performance only in few cases: if the development and test data are similar w.r.t. sentence length distribution. If this is not given, the separate thresholds cannot be estimated reliably on the development data.
- For the system-based confidence measures, there is no significant difference between the measures computed over the word graph and those over N-best lists for WER based classification. An N-best list of length 10,000 is sufficient for estimating the word posterior probabilities reliably. Longer lists do not further improve the confidence measure.

7.3 Rescoring with confidence measures

This section reports on the use of word posterior probabilities for rescoring. N-best lists generated by the RWTH SMT system described in section 3.3.3.1 have been used as input.

The rescoring is performed as follows: For every hypothesis in the N-best list, the confidence of each word in the sentence is calculated. These word posterior probabilities are multiplied to obtain a score for the whole sentence:

$$Q_{prod}(e_1^I, f_1^J) = \prod_{i=1}^{I} p(e_i \mid f_1^J) \; . \tag{7.1}$$

This sentence score is then used as additional model for N-best list rescoring. It serves as an indicator of the overall quality of the generated hypothesis. Additionally, the minimal word posterior probability over the sentence is determined. This can be seen as an indicator whether the hypothesis contains words which are likely to be incorrect:

$$Q_{min}(e_1^I, f_1^J) = \min_{i=1,\ldots,I} p(e_i \mid f_1^J) \; . \tag{7.2}$$

Rescoring has been carried out on EPPS data using the direct phrase-based confidence measures.

Within the project TC-STAR, an MT evaluation campaign was performed in March 2005 to compare the research systems of the consortium members. Different conditions concerning the input data were defined, which are explained in appendix A.2. In the following, rescoring results on the verbatim transcriptions will be presented.

RWTH submitted translations from the phrase-based translation system described in section 3.3.3.1 to this evaluation. N-best lists were generated for development and test corpus, with a maximal length of 20,000 and 15,000, respectively. These were then rescored with an IBM model 1, a fourgram language model, and a deletion model based on IBM-1. The weights for all these models and for the sentence probability assigned by the SMT system were optimized w.r.t. BLEU score on the development corpus. For a detailed description of the system, see [Vilar et al. 05]. This system was ranked first in the evaluation round according to all evaluation criteria [Ney et al. 05]. At this early stage of the project, only automatic evaluation using WER, PER, BLEU and NIST was performed. Human assessment will be carried out in later evaluation rounds.

Two different sets of experiments have been carried out for this thesis. In both cases, rescoring with the direct phrase-based word posterior probabilities has been performed as described in equations 7.1 and 7.2. The sub-model weights have been optimized w.r.t. BLEU score. The two investigated settings are the following:

1. Start from the baseline system without rescoring. The sub-model weights of this system have been optimized w.r.t. BLEU on the development set, but no additional models have been used for rescoring the N-best list. This experiment has been performed to analyze the maximal improvement which can be achieved through rescoring with confidence measures.

2. Start from the system which has already been rescored with the three models mentioned above. This is the system which was used in the TC-STAR evaluation campaign, and which was ranked first. In this setting, it can be seen whether the rescoring with confidence measures manages to improve over the best available system as well. Furthermore, it can be analyzed whether the gains from all rescoring models are additive.

The experimental results are shown in table 7.24. The upper block contains the evaluation of translation quality without regarding case, and the lower block shows the case-sensitive evaluation results. These different figures are presented here because the translation system has been trained on a lower-cased corpus. The true-casing is performed as an additional post-processing step. Thus, the effect of translation and rescoring can be separated from that of the true-casing process.

First, the case-insensitive results will be discussed. The baseline is the single best output of the translation system. This system can be improved through rescoring with confidence measures by 1 BLEU point. This is only 0.1 BLEU points less than the gain achieved from rescoring with the three other models. In terms of the other evaluation measures, there is also an improvement, even though it is not as large. This does not come unexpected since the system has been optimized w.r.t. BLEU. Now consider the translations which have been submitted to the evaluation campaign. The rescoring with IBM model 1, the language and the deletion model yields an improvement of 1.1 BLEU points over the single best baseline. Another 0.6 BLEU points can be gained through additional rescoring with the direct phrase-based confidence measures. The improvement is consistent across all four automatic evaluation criteria.

Table 7.24: Translation quality for rescoring with confidence measures. EPPS Spanish → English test set, version: verbatim. Optimized for BLEU. Case insensitive and case sensitive evaluation.

case	system	WER[%]	PER[%]	BLEU[%]	NIST
no	baseline	40.9	30.4	45.5	9.83
	+ direct phrase-based conf. measures	40.8	29.9	46.5	9.93
	+IBM-1+LM+deletion model *	40.6	29.5	46.6	9.99
	+direct phrase-based conf. measures	**40.4**	**29.4**	**47.2**	**10.04**
yes	baseline	42.5	32.2	45.1	9.67
	+ direct phrase-based conf. measures	42.7	32.0	45.6	9.68
	+IBM-1+LM+deletion model *	42.5	31.7	45.9	9.75
	+direct phrase-based conf. measures	**42.4**	**31.6**	**46.2**	**9.78**
	second best translation system	43.9	33.4	44.1	9.47

* : official RWTH submission in the TC-STAR evaluation campaign

In the TC-STAR evaluation campaign, case information was taken into account. The corresponding results are presented in the lower block of the table. The overall translation quality is lower if case is considered. For all models applied here, the gain achieved through rescoring is not as big as in the case insensitive evaluation. If the confidence measures only are used for rescoring, there is no consistent effect: BLEU score is slightly higher, whereas NIST score and the error measures deteriorate slightly. However, when all four rescoring models are applied, the system is significantly improved. The models used in the TC-STAR evaluation yield an increase of 0.8 BLEU points. The word posterior probabilities add another 0.3 points to this. This change is rather small, but comparable to the contribution of each single rescoring model used in the evaluation campaign. For comparison, the translation quality of the second best system in this campaign is reported in the last row of the table. The BLEU score of the RWTH system can be significantly improved through rescoring so that the difference to the second best system gets clearer. The distance increases from 1.0 to 2.1 BLEU points. A complete overview of the results of the evaluation campaign is given in [Ney et al. 05].

7.4 Application in interactive statistical machine translation

In this section, results for the application of confidence measures in an interactive translation system will be presented. This system has been developed in the European project TransType2. For a description of the interactive MT system, see section 6.

7.4.1 Experimental setting

The experiments have been performed on two corpora from the TT2 Xerox collection. The translation directions are French → English and English → German; see tables A.1 and A.2 in appendix A for the corpus statistics. The statistical machine translation system has been trained and tuned on these corpora. The lexicon used to estimate the word confidences has been trained on the same data. The interactive alignment templates system has been used for the translation prediction experiments. The test corpora have been translated in simulated interactive mode, which is described in section 6.3.1.

The results are evaluated using the metrics described in section 6.3.2: recall[b] and precision on the proposed extensions, as well as their combination into the prediction F-measure F_p. Additionally, the number of proposed extensions has been counted as a measure for the user–system interaction. First, the results for confidence based rejection and selection of words will be analyzed on the French → English task. Then, it will be shown how this can be improved by exploring information from the given prefix. Finally, results achieved with the best system setup on a different language pair, namely English-German, will be presented.

7.4.2 Effect of confidence based rejection and selection of words

When applying the confidence based selection and rejection of words, experiments have shown that the system performs best if

- the selection of words on the word graph (as described in step 2 in section 6.1) is guided only by the score assigned by the statistical machine translation system,
- both the confidence of the word and the score assigned by the SMT system are considered in the language model based prediction (as described in step 3),
- the rejection of words is performed both in step 2 and 3 of the search.

This setting has been used in all experiments reported in this thesis.

Figure 7.9 shows the effect of confidence based selection and rejection of words on precision, recall and F_p for different values of the confidence scaling factor and the confidence threshold. From left to right, the figures plot recall, precision and F_p versus the confidence threshold. A confidence threshold of 0 corresponds to accepting all words, and setting it to 1 means that all words are rejected. If the scaling factor is 0, the confidence estimation is not used for selection of words. Experiments have been run with several non-zero confidence scaling factors. The best results have been obtained for a scaling factor of 0.8, but the prediction performance is very similar for all scaling factors between 0.5 and 1. Thus, only results for the optimal scaling factor value 0.8 and the baseline (factor 0) are presented here. As can be seen in the left plot in figure 7.9, recall decreases as the confidence threshold is increased – which is to be expected because the predictions proposed by the system get shorter and sometimes correct words are discarded. On the

[b] Recall is defined as 1−KSR.

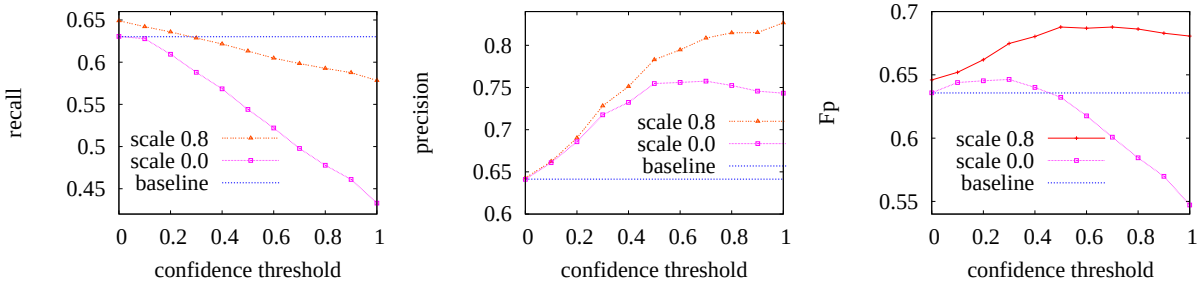


Figure 7.9: Effect of selection and rejection of words on recall, precision and F_p (from left to right) for different threshold values on the French → English test set. The values of the confidence scaling factor are 0 (baseline) and 0.8.

other hand, precision rises substantially as the right-next plot shows. When comparing the systems with confidence scaling factors 0 and 0.8, one can see that both recall and precision are higher if the selection of words is based on their confidence, i.e. if the scaling factor is set to 0.8. This causes the harmonic mean F_p to increase significantly over the baseline. The best result with F_p= 68.9% is obtained for a confidence scaling factor of 0.8 and a threshold value of 0.5.

When analyzing the plots in figure 7.9, it seems surprising that precision and recall are never zero. This is the case even if the confidence threshold is set to 1, i.e. if all words are rejected. Due to the setup of the language model predictor (see step 3 on page 65), the system will always predict the word with the highest language model probability if all words in the word graph are discarded. This causes precision and recall to be always greater than zero.

Examples

Figure 7.10 shows examples of sentences from the French-English test corpus that have been translated in simulated interactive mode. The different extensions proposed by the system with and without confidence based rejection are given. The accepted prefix which is taken into account is shown in grey. The value of the confidence scaling factor is set to 0.8 which is optimal with respect to F_p. The part of the translation that is accepted by the user, together with the following character that he or she enters, is taken as prefix for the new search. For each sentence, first the necessary steps are shown for a system which does not discard any of the proposed words. These are the results obtained with a confidence threshold of 0. Then, the same is presented for a a threshold of 0.5. In the first example, the correct result is obtained much faster with confidence based rejection than without: only four steps of user–system interaction are necessary instead of seven. Furthermore, bad translations like the word "Remote" and "for" are discarded and not proposed to the user at all. The improvement of the system's performance is also quite clear in terms of precision and recall: The number of keystrokes necessary to type the reference decreases from 11 to 5; and precision rises drastically from 55.0% to 82.9%. So, the gain for the user is substantial; and the quality of the proposed translations increases noteworthily.

source			Utilitaire d'impression pour Macintosh vi
reference			Macintosh Printer Utility vi
confidence threshold	0	prefix	
		extension	Remote Print for Macintosh vi
		prefix	M
		extension	Management print for Macintosh vi
		prefix	Mac
		extension	Macintosh vi utility to print
		prefix	Macintosh P
		extension	Macintosh Print utility to vi
		prefix	Macintosh Printe
		extension	Macintosh Printer
		prefix	Macintosh Printer U
		extension	Macintosh Printer U tility to vi
		prefix	Macintosh Printer Utility v
		extension	Macintosh Printer Utility vi
confidence threshold	0.5	prefix	
		extension	Macintosh vi to
		prefix	Macintosh P
		extension	Macintosh Printer
		prefix	Macintosh Printer U
		extension	Macintosh Printer Utility
		prefix	Macintosh Printer Utility v
		extension	Macintosh Printer Utility vi
source			Répertoire téléphonique vi
reference			Fax Phonebook vi
confidence threshold	≤0.3	prefix	
		extension	Fax Phonebook vi
confidence threshold	0.5	prefix	
		extension	
		prefix	F
		extension	Fax
		prefix	Fax P
		extension	Fax Phonebook
		prefix	Fax Phonebook v
		extension	Fax Phonebook vi

Figure 7.10: Effect of rejection and selection on the proposed extensions for different values of the confidence threshold. Illustration by examples from the TT2 Xerox French → English test corpus; simulated interactive mode. The confidence scaling factor is set to 0.8.

Table 7.25: Average extension length (in characters) per extension and number of proposed extensions for different confidence thresholds on the TT2 Xerox French → English test set. The confidence scaling factor is set to 0.8.

confidence threshold	baseline	0.0	0.1	0.2	0.3	0.5	0.7	1.0
avg. length / source sentence	238.9	234.0	199.8	170.1	140.2	104.4	84.3	61.9
avg. length / extension	14.6	15.2	12.6	10.5	8.4	5.9	4.5	3.0
avg. # extensions / sentence	16.3	15.4	15.8	16.2	16.7	17.8	18.9	20.7

The second example on the other hand shows that the rejection of words can also harm performance: For low threshold values (≤ 0.3), the system proposes the correct translation immediately. If the confidence threshold is set higher, correct words are discarded. Three additional steps of user–system interaction are necessary. The precision is not affected in this case, but the recall decreases.

Extension lengths and number of extensions

As the examples above show, the extensions proposed by the system with confidence estimation are more accurate, but also significantly shorter. A detailed analysis of this effect is given in table 7.25: It presents the average length of the proposed extensions per source sentence and per extension on the test set. This length drops significantly as more and more translations are discarded. For a threshold of 1, only three characters per extension are proposed on average. In order to obtain predictions that are not too short, the value for the confidence threshold should not be set too high. For a new domain or a new language pair, it might not always be clear what is a good choice for the confidence threshold. One way to account for this is the adaptation of the threshold as described in section 6.2.3. Results for this adaptation method will be presented in the following subsection.

The last row of table 7.25 shows the average number of proposed extensions per source sentence. For thresholds below 0.3, the number of user–system interactions is reduced compared to the baseline. For higher thresholds, this number increases significantly. It seems thus advisable to set the threshold relatively low and not to discard too many hypotheses.

7.4.3 Effect of using prefix information

Three different sets of experiments have been performed to study the effect of prefix information on the quality of the predicted extensions. The methods proposed in section 6.2.3 are applied as follows:

1. the confidence calculation is restricted to the untranslated source words as introduced in equation 6.2; this will be labeled as “source” in the tables and figures,

Table 7.26: Effect of using prefix information for confidence threshold or source sentence restriction on recall [%] and precision [%] for different values of the confidence threshold on the TT2 Xerox French → English test set. The confidence scaling factor is set to 0.8.

confidence threshold			0.0	0.1	0.3	0.5	0.7	1.0
recall [%]	use prefix:	no	64.9	64.2	62.9	61.3	59.8	57.8
		source	64.9	64.2	62.9	61.4	60.1	58.2
		threshold	64.9	64.2	63.0	61.7	60.6	59.1
precision [%]	use prefix:	no	64.3	66.2	72.8	78.3	80.9	82.7
		source	64.3	67.3	74.1	79.1	81.3	82.9
		threshold	64.3	66.1	71.7	76.2	79.0	80.9

2. the confidence threshold is adapted as described in section 6.3; labeled as "threshold",

3. the methods 1 and 2 are combined.

These approaches have been compared to an interactive system applying confidence estimation without exploring the prefix information.

Table 7.26 shows an assessment of prediction quality in terms of precision and recall for the modified confidence estimation (method 1) and for the adaptation of the confidence threshold (method 2). When only adapting the confidence threshold, the recall of the proposed extension increases, especially for higher threshold values. This is due to the fact that fewer correct words get discarded. But on the other hand, precision decreases significantly. Confidence estimation based on a restricted source sentence, on the other hand, yields a slight degradation of recall. But there exists a significant improvement in terms of precision, especially for low threshold values. The reason for this is that translations of source words which have been covered already by the prefix will now be correctly discarded.

The effect of all three methods presented above on F_p is shown in figure 7.11. The calculation of word confidences according to a restricted source sentence ("source") consistently improves the predictions: The F-measure F_p increases over the system without use of prefix information for all threshold values. Additionally, the adaptation of the confidence threshold can compensate for the negative effect of setting the threshold too high: For threshold values above 0.7, the system with threshold adaptation performs better than that without. The combination of these two methods achieves the best results in terms of F_p for threshold values of 0.7 and higher. For lower threshold values, the restriction of the confidence calculation to the uncovered parts of the source sentence performs best. In general, the latter method of using prefix information should be preferred.

An example of system output with and without consideration of the given prefix for confidence estimation is given in table 7.27. The words which have not been translated yet are printed in italic. The system that does not take the prefix into account proposes the word "System" as next extension, although the source word "système" has been translated

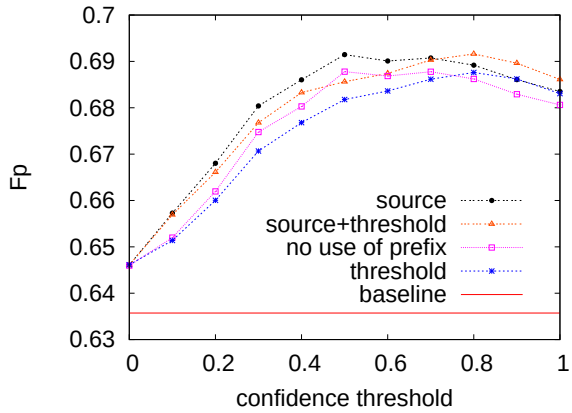


Figure 7.11: Effect of using prefix information for confidence threshold and source sentence restriction on F_p for different threshold values on the TT2 Xerox French → English test set. The confidence scaling factor is set to 0.8.

Table 7.27: Effect of calculating confidence over a restricted source sentence on the proposed extensions. Example from the TT2 Xerox French → English test set, simulated interactive mode. The confidence scaling factor is set to 0.8, and the confidence threshold is 0.5.

source	Reportez-vous au chapitre 9 - *Réglage du* système
reference	Refer to Chapter 9 - System Setup
prefix	Refer to Chapter 9 - System S
system without prefix information	ystem
system with prefix information	etup

already. The reason for this is that it has a high confidence w.r.t. the complete input sentence whereas the correct target word "Setup" has not. The improved system that determines the confidence only over the uncovered source words "Réglage du" is able to predict the correct extension "Setup".

7.4.4 Comparative results on English-German

In order to verify the gain in the interactive system's performance, additional experiments have been run on the translation from English into German which is different from French → English in terms of structure and complexity. This language pair has been also investigated in TransType2. The system which proved best in the French → English experiments has been used, i.e.

- selection of words in search step 3 (see page 65 in section 3),
- rejection of unconfident words in all search steps,

- using the knowledge contained in the correct prefix for adaptation of the confidence estimation (see equation 6.2 in section 6.2.3).

The performance of this system on the English → German test set in terms of F_p is shown in the left plot in figure 7.12. The gain in performance is even higher on this translation task than for French → English: The system improves by 13.4% absolute over the baseline. When analyzing the number of user–system interactions presented in the right plot in figure 7.12, one sees a significant reduction over the baseline for almost all threshold values. This can be explained by the fact that on the German → English translation task, the baseline system proposes completions of lower quality than on French → English. The gain from using confidence estimation for discarding words is thus much higher on this language pair.

7.4.5 Conclusion

Several novel ways of applying word-level confidence measures in an interactive statistical machine translation system have been presented and experimentally evaluated. The system is a state-of-the-art statistical machine translation system that suggests translations and adjusts them to a prefix that has already been typed by a user. The results can be summarized as follows:

- The correctness of the translations predicted in this process can be significantly improved through rejection and selection of words based on their confidence.
- The confidence based rejection of words increases precision, whereas the selection has a positive effect on both recall and precision.
- The quality of the predictions can be improved by calculating the word confidences only over the part of the source sentence which has not been translated.
- The gain in prediction performance is 5.6% absolute over the baseline on the French → English test corpus, and 13.4% absolute on the English → German test set.

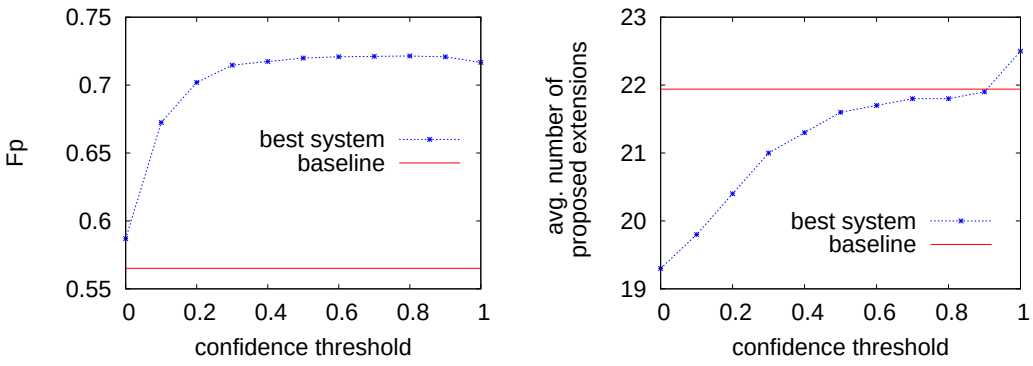


Figure 7.12: F_p and average number of proposed extensions for the best system compared to the baseline system on the TT2 Xerox English → German test set. The confidence scaling factor is set to 0.8.

- The number of user–system interactions decreases by up to 12 % relative through the use of confidence measures.

8 Scientific contributions

The aim of this thesis was to set up a probabilistic framework for the computation of word posterior probabilities for machine translation. Within this framework, different concepts of word posterior probabilities were defined and analyzed. Several approaches to the calculation of word posterior probabilities were investigated and compared. The use of word posterior probabilities as confidence measures was studied, including their application in different scenarios. In particular, the following aspects were studied:

Word posterior probabilities

For the computation of word posterior probabilities, several different conditions were defined under which a word can occur in a sentence. They include a definition based on the target position of the word, its frequency in the sentence, and the occurrence in any target position. Different approaches to the calculation of these word posterior probabilities were introduced: the system-based and the direct approach. The system-based approach derives the word posterior probability from the sentence posterior. To this end, all translation hypotheses contained in a word graph or in an N-best list are considered, and their probabilities are summed to obtain the word posterior probability. Efficient methods for performing this summation were presented. The direct methods take external knowledge sources – such as word or phrase lexicon models – into account. They are independent of the translation system which generated the translation. The methods introduced in this thesis are based on HMMs and on phrase-based translation models.

Additionally, several aspects of the computation of word posterior probabilities were investigated. They include the scaling of the probability density functions for confidence estimation, and normalization issues for the direct phrase-based approach.

Confidence measures

The use of the word posterior probabilities proposed in this thesis as word-level confidence measures was investigated. Several ways of determining the correctness of words in a translation were discussed. They are derived from different well-known MT evaluation measures. Further aspects which were studied include: the definition of the acceptance threshold, the relation between the confidence measure and the word error measure which defines the reference tags, as well as optimization of the confidence measures for the classification task. All experiments were evaluated with two different, well-established metrics. All confidence measures in this thesis were systematically evaluated on several different translation tasks and different language pairs.

On all corpora, the best methods proposed here reduce the confidence error rate significantly (at the 1%-level). The direct confidence measures were also successfully applied to output from a non-statistical MT system. Moreover, the word posterior probabilities introduced in this thesis proved to be the best performing single features in confidence estimation experiments carried out in the CLSP summer workshop 2003. The combination of several variants of word posterior probabilities into one confidence measure was investigated on several tasks and language pairs. The resulting confidence measure yields an improvement over the best stand-alone feature.

Improving translation quality through confidence estimation

The confidence measures introduced in this thesis were used to improve translation quality in two different ways: They were integrated into an interactive SMT system, and they were used for rescoring over N-best lists. The interactive system is based on a state-of-the-art approach to SMT. Novel ways of exploring confidence estimation in this system were presented. A new metric for measuring the quality of an interactive system in offline mode was developed. It combines the established recall-based keystroke ratio with a component measuring precision. It was shown that the prediction quality of the system can be improved through the use of confidence measures.

Additionally, rescoring with confidence measures was shown to improve translation quality. The investigated SMT system was the one which was ranked first in the TC-STAR evaluation campaign in March 2005. It was consistently improved through rescoring with confidence measures.

Connection with Bayes decision rule

In this thesis, the loss functions for several MT word error measures were formulated. After deriving the posterior risk for these loss functions, it was shown that the developed word posterior probabilities are a major component of the risk. This yields a theoretical foundation for the different ways of defining word posterior probabilities. The experimental evaluation showed that the word posterior probabilities perform very well for confidence estimation if the reference tags are defined by the same word error measure.

9 Future directions

The word posterior probabilities for machine translation proposed in this thesis have been applied as confidence measures and experimentally evaluated. Additionally, applications improving translation quality have been studied. Interesting research topics following these lines include several approaches which explore the information about confidence in the translation process.

The application of confidence measures in rescoring could be studied in more detail. Methods which filter the translation hypotheses in the N-best list or recombine them to generate new translations are imaginable. A related research topic is the use of word confidence measures in system combination. In this approach, output from different machine translation systems is combined to create new translations. This method has recently been proposed and shown to be successful. Confidence measures could be applied to select or discard words in this combination.

More generally speaking, the information provided by confidence measures could be used by natural language processing systems which take the automatically generated translations as input. Examples include information extraction or question answering systems.

The application of confidence measures in interactive machine translation could be investigated further, tackling issues like the use of more sophisticated confidence measures in the system. The rather simple confidence measure based on IBM model 1 could be replaced by the direct phrase-based approach, for example. Moreover, refined ways of exploring the information from the correct prefix for confidence estimation deserve consideration.

A Corpora

This appendix contains a description of the different corpora used for the experiments in this thesis. Translations on three different tasks have been investigated: on the Xerox data from the EU project TransType2, on speeches of the European parliament collected within the joint European project TC-STAR, and on Chinese new articles from the MT evaluation campaign carried out by the National Institute of Standards and Technology (NIST).

A.1 TransType2 Xerox

The aim of the EU project TransType2 was to develop a computer-assisted translation system, which helps to meet the growing demand for high-quality translation. The project lasted from March 2002 to February 2005. The approach pursued within TransType2 is to embed a statistical machine translation engine with an interactive translation environment. Such an interactive SMT system is described in section 6.1. Six different versions of the system have been developed for French, Spanish and German, each translated from and into English. To ensure that TransType2 corresponds to the translators' needs, two professional translation agencies have evaluated successive prototypes. For more details about the project, see [tt2 05].

The so-called Xerox corpora compiled within the project consist of technical manuals like user guides, operating guides, and system administration guides for Xerox devices such as copiers and printers. The domain is restricted and has a very specialized terminology. This task is typical of work that is done by human translators in translation agencies.

The corpus statistics for training, development and test sets in the three different language pairs are given in tables A.1 through A.3. All corpora exist in different versions: the so-called "simplified" version, and the "raw" version. For the simplified version, the corpora have been preprocessed in order to reduce vocabulary size and facilitate translation. Preprocessing steps include categorization of numbers, punctuation marks and special characters, and lowercasing of the text.

The experiments reported in this thesis have been performed on all three language pairs, where the raw versions of French $\leftrightarrow$ English and German $\leftrightarrow$ English and the simplified version of Spanish $\leftrightarrow$ English have been used. Comparing the corpus statistics of the raw corpora (tables A.1 and A.2) with those of the simplified version (table A.3), a large reduction of vocabulary size and number of singletons and unknown words (OOVs) is noticeable.

Table A.1: Statistics of the French-English training, development, and test sets. TransType2 Xerox corpus, technical manuals, version: raw.

		French	English
TRAIN	sentences	53 046	
	running words	680 796	628 329
	running words without punctuation marks	627 027	573 912
	vocabulary	15 632	13 816
	singletons	4 789	4 032
DEV	sentences	994	
	running words	11 674	10 903
	running words without punctuation marks	10 700	9 917
	number of distinct words	1 895	1 795
	OOVs (running words)	184	141
TEST	sentences	984	
	running words	11 709	11 177
	running words without punctuation marks	10 889	10 358
	OOVs (running words)	204	201

Table A.2: Statistics of the German-English training, development, and test sets. TransType2 Xerox corpus, technical manuals, version: raw.

		English	German
TRAIN	sentences	49 376	
	running words	589 531	537 464
	running words without punctuation marks	509 902	443 547
	vocabulary	13 223	23 845
	singletons	3 681	9 443
DEV	sentences	964	
	running words	10 642	10 462
	running words without punctuation marks	8 372	9 259
	number of distinct words	1 516	1 746
	OOVs (running words)	29	147
TEST	sentences	996	
	running words	11 704	12 298
	running words without punctuation marks	9 711	10 656
	vocabulary	2 179	1 838
	OOVs (running words)	485	142

Table A.3: Statistics of the Spanish-English training, development, and test sets. TransType2 Xerox corpus, technical manuals, version: simplified.

		Spanish	English
TRAIN	sentences	55 761	
	running words	752 606	665 399
	running words without punctuation marks	688 498	602 224
	vocabulary	11 050	7 956
	singletons	3 156	1 928
DEV	sentences	1 012	
	running words	15 957	14 278
	running words without punctuation marks	14 689	12 972
	number of distinct words	1 224	1 433
	OOVs (running words)	54	27
TEST	sentences	1 125	
	running words	10 106	8 370
	running words without punctuation marks	9 589	7 873
	number of distinct words	1 215	1 132
	OOVs (running words)	69	49

A.2 TC-STAR EPPS

The EPPS corpora were compiled in the TC-STAR project. This is an ongoing EU project which started in April 2004. Its aim is to improve core technologies for speech translation in unconstrained domains. The EPPS corpora comprise speeches from the plenary sessions of the European Parliament.

The European Parliament usually holds plenary sessions six days each month. The major part of the sessions takes place in Strasbourg (France) while the residual sessions are held in Brussels (Belgium). Today the European Parliament consists of members from 25 countries, and 20 official languages are spoken. The sessions are chaired by the President of the European Parliament. Simultaneous translations of the original speech are provided by interpreters in all official languages of the EU.

The European Union's TV news agency, *Europe by Satellite (EbS)*[a] broadcasts the European Parliament Plenary Sessions (EPPS) live in the original language and the simultaneous translations via satellite on different audio channels: one channel for each official language of the EU and an extra channel for the original untranslated speeches. These channels are also available as 30-minute long internet streams for one week after the session.

A preliminary version, the so-called verbatim transcripts, of the texts of the speeches

[a] `http://europa.eu.int/comm/ebs/`

given by members of the European Parliament are published on the EUROPARL website[b] on the day after the session. After a time period (about two months) in which the politicians are allowed to make corrections the speeches are available in their final form, known as Final Text Edition, in all official languages of the EU. The website also provides all previous reports since April 1996. The available reports have been used for building an English-Spanish parallel text corpus for the TC-STAR project.

Before training the translation models, some preprocessing steps have been carried out in order to better adapt the training material to the evaluation conditions. In the texts to be translated, some normalization has been performed, like expanding contractions ("I am" instead of "I'm", "we will" instead of "we'll", etc.) and eliminating hesitations ("uhm-", "ah-", etc.). In addition, as current state-of-the-art speech recognition systems do not generate (reliable) punctuation marks and most of them produce only lowercased text, the punctuation marks of the training texts have been eliminated. Additionally, all numbers are written out (e.g. "forty-two" instead of "42").

The statistics of the corpora used for training, development and testing can be seen in table A.4. The development and test corpora are both provided with two references on the English side. The test texts correspond to the plenary sessions held between the 15th and 18th November 2004. It should be noted that the notion of "sentences" is oriented towards automatic speech recognition. For the verbatim transcriptions the "sentences" correspond to the segmentation used for the speech recognition systems. This is guided by silence in the audio signal more than by grammatical constructs.

Table A.4: Statistics of the TC-STAR EPPS Spanish-English training, development, and test sets. Version: verbatim. Punctuation marks have been removed from the corpus. Both development and test corpus are provided with 2 English references.

		Spanish	English
Train	sentences	1 652 174	
	running words	32 554 077	31 147 901
	vocabulary	124 192	80 125
	singletons	41 148	27 631
Dev	sentences	2 643	
	running words	20 289	40 396
	number of distinct words	2 932	3 051
	OOVs (running words)	46	499
Test	sentences	1 073	
	running words	18 896	37 742
	number of distinct words	3 302	3 321
	OOVs (running words)	145	172

[b] http://www.europarl.eu.int/plenary/

A.3 NIST Chinese-English

Two different sets of experiments have been performed on the Chinese-English data from the NIST MT evaluation: The methods proposed in this thesis have been applied to output from the RWTH phrase-based translation system on the NIST Chinese → English translation task. In the CLSP summer workshop 2003, translations generated by the ISI alignment templates system were used as input for confidence estimation. Both experimental settings will be described in this section.

NIST MT evaluations

Starting in 2001, NIST has carried out yearly MT evaluation rounds for Chinese → English translation [NIST 02]. The domain is news articles, covering a wide range of different topics. The goal of the evaluation is to set up a framework for objectively comparing different MT systems. Through this, progress is to be promoted for MT technologies. All submissions are evaluated automatically using different translation evaluation measures. In addition, a part of the submitted translations is evaluated manually.

The training data is explicitly specified. The participating groups are not allowed to use any other data for developing their systems. The corpora are taken from the following sources:

- news from several agencies, newspapers and magazines,
- the record of the proceedings (Hansards) from the Hong Kong Legislative Council,
- bilingual text from the United Nations website.

In addition, a bilingual conventional dictionary, provided by LDC, and English news texts for language model training are used as resources.

Translations from the RWTH system

Translations generated by the RWTH phrase-based system have been used to evaluate the confidence measures proposed in this thesis. The SMT system has been trained on the corpora described in table A.5. Both monolingual and bilingual data have been used in training. The NIST evaluation set from 2002 serves as development set both for the SMT system and for the confidence measures. The test data from 2004 has been used for assessing the quality of the confidence estimation. Both sets are provided with four English references. N-best lists with a maximal length of 10,000 have been generated for development and test set.

Translations from the ISI system

In the CLSP summer workshop on confidence estimation for machine translation, the team worked on data from the NIST 2001 and 2002 MT evaluation rounds. The translations

Table A.5: Statistics of the NIST Chinese-English corpora. Mono- and bilingual data are used for training, including a conventional dictionary. The development corpus is the 2002 evaluation set. The test corpus is the 2004 evaluation set. Both development and test corpus are provided with 4 English references.

			Chinese	English
TRAIN	bilingual data	sentences	7M	
		running words	199M	213M
		vocabulary	223K	351K
		dictionary entries	82K	
	monolingual data	sentences	-	20M
		running words	-	636M
		vocabulary	-	351K
	LM perplexities	trigram	-	124
		fourgram	-	105
DEV (eval 2002)		sentences	878	
		running words	25K	105K
TEST (eval 2004)		sentences	1 788	
		running Words	52K	239K

were generated by the alignment templates system from ISI. This system is similar to the RWTH alignment templates system. It was trained on the data from the NIST evaluation campaign.

The data used for training and evaluating the confidence measures are the automatic translations generated on two collections of Chinese source sentences:

1. 993 sentences from the evaluation set of the 2001 NIST MT evaluation campaign, each with 4 reference translations,
2. 878 sentences from the evaluation set of the 2002 NIST MT evaluation, each with 4 reference translations.

1,000-best lists for both sets have been generated using the ISI alignment templates system. The available data was split into three datasets used to carry out the machine learning experiments:

- The *training set* was used to estimate the parameters of the confidence estimation models which combine different features. It consists of the translations of the first 700 source sentences from the NIST 2001 evaluation set.
- The *validation* or *development set* was used to tune some hyper-parameters of the learning process, such as the model structure, number of iterations of the training procedure, etc. The remaining 293 N-best lists from the first collection were used as this validation set.

- The *test set* was used to evaluate the final performance of the model, after training and optimization. The test set consists of the 878 N-best lists from the NIST 2002 evaluation data.

The statistics for training, development and test corpus are given in table A.6. Note that these are the corpora used to train the confidence estimation models, and not the statistical machine translation system.

Table A.6: Statistics of the NIST Chinese-English training, development, and test sets used in the CLSP summer workshop 2003. 1,000-best lists generated by the ISI alignment templates system.

	sentences		running Words
	Chinese	English	English
TRAIN	700	698 082	20 736 971
DEV	293	292 870	7 492 753
TEST	878	876 831	26 360 766

B MT evaluation measures

This appendix explains the different evaluation metrics used for assessment of translation quality in this thesis. For a detailed description, see for example [Leusch et al. 05].

WER (**W**ord **e**rror **r**ate):
The word error rate (WER) is based on the Levenshtein distance [Levenshtein 66]. It is calculated as the minimum number of substitutions, deletions and insertions that have to be performed in order to transform the translation hypothesis into the reference sentence. If there exist multiple references for each sentence, the Levenshtein distance to the most similar reference is calculated [Nießen et al. 00].

PER (**P**osition-independent word **e**rror **r**ate):
The word order of two target sentences can be different even though they are both correct translations. To account for this, the position-independent word error rate proposed by [Tillmann et al. 97] compares the words in the two sentences *without* taking the word order into account. The PER is always lower than or equal to the WER.

BLEU (**Bi**lingual **e**valuation **u**nderstudy):
This evaluation measure has been proposed by [Papineni et al. 02]. It is based on the notion of modified n-gram precision, with $n \in \{1, \ldots, 4\}$: All candidate unigram, bigram, trigram and fourgram counts are collected and clipped against their corresponding maximum reference counts. The reference n-gram counts are calculated on a corpus of reference translations for each input sentence. The clipped candidate counts are summed and normalized by the total number of candidate n-grams. The geometric mean of the modified precision scores for a test corpus is calculated. This is then multiplied with an exponential brevity penalty factor to penalize translations which are too short. BLEU is an accuracy measure.

NIST (NIST score):
[Doddington 02] enhanced the BLEU score into the NIST measure by taking into account n-gram information weights, and using a different n-gram averaging scheme and a different brevity penalty. The NIST measure weights each n-gram by the information gain of the n-gram itself and its $(n-1)$-prefix. Like BLEU, it is an accuracy measure.

Evaluation metrics for interactive MT are described in section 6.3.2. The state of the art is described there, and a new evaluation metric is proposed which allows for a more detailed analysis of the system's performance.

C Statistical machine translation systems

This appendix describes the SMT systems which have been used for the experiments on confidence estimation presented in this thesis. The RWTH translation systems based on alignment templates and on finite state transducers will be presented here. The RWTH phrase-based translation system is depicted in more detail in section 3.3.3.1 because this is the basis for the confidence measures introduced there.

C.1 Alignment templates approach (AT)

The key elements of the translation approach described in this section are so-called *alignment templates* [Och and Ney 04]. These are pairs of source and target language phrases with an alignment within the phrases. The alignment templates are build at the level of word classes. This improves the generalization capability of the model.

The system takes a number of different sub-models into account which are combined log-linearly. Those sub-models are:

- a phrase translation model,
- a word translation model,
- two language models: a word-based n-gram model and a class-based n-gram model,
- two heuristics: the word penalty and the alignment template penalty which assign costs to each word and each alignment template that is generated,
- three feature functions that model reordering on the level of alignment templates and on the word-level.

A dynamic programming beam search algorithm is used to generate the translation hypothesis with maximum probability for a given source sentence. This search algorithm allows for arbitrary reorderings at the level of alignment templates. Within the alignment templates, the reordering is learned in training and kept fixed during the search process. There are no constraints on the reorderings within the alignment templates.

This is only a brief description of the alignment template approach. For further details, see [Bender et al. 04, Och and Ney 04].

C.2 Finite state transducer approach (FSA)

Statistical machine translation may be viewed as a weighted language transduction problem [Vidal 97]. This makes it possible to build a machine translation system with the use of weighted finite-state transducers. The translation problem can be solved by estimating a language model on a bilanguage defined over source and target language (see also [Bangalore and Riccardi 00, Casacuberta et al. 01]). This bilanguage is learned from the training corpus that has been automatically word aligned beforehand. The following shows two examples of this bilanguage for a Catalan-English translation task:

```
podria|could_you donar|$ per|$ favor|please_give_me el|$ seu|your
      numero|number ?|?

si|yes ,|, be|well ,|, no|I_do_not se|know .|.
```

Note that Catalan is simplified and does not contain accents. The bilanguage words are of the form `source|target`. The symbol `$` denotes transitions that contain a source but no target word. Those can occur if the source word does not have any correspondence in the target sentence. Another reason can be that, due to the fixed segmentation given by the word alignments, phrases in the target language are moved to the last source word of an alignment block, as the example "`donar per favor`" / "`please give me`" shows.

The steps for training the FSA translation system are the following:

1. Perform word alignment (using the publically available GIZA++ toolkit[a]),
2. transform the training corpus with a given alignment into the corresponding bilingual corpus,
3. train a language model on the bilingual corpus,
4. build an acceptor A from the language model.

Then the translation problem from above can be solved by composition of finite state transducers. A more detailed description of the system can be found in [Kanthak and Ney 04].

[a] `http://www-i6.informatik.rwth-aachen.de/web/Software/index.html`

D Additional experimental results

In this appendix, additional experimental results will be presented. Section D.1 will describe experiments in which the confidence measures proposed in this thesis are evaluated on output from four different MT systems on TT2 data. This rounds off the results shown in section 7.2.3. A detailed data analysis comparing the Spanish-English data from the TransType2 and the EPPS task will be shown in section D.2. This completes the comparison between the system-based and the direct phrase-based confidence measures given in section 7.2.6. The distribution of sentence lengths on four different corpora will be shown in section D.2.2. This analysis shows how much gain can be expected from the alternative definition of acceptance thresholds as proposed in section 4.5.

D.1 Additional classification results

This section presents additional results from experiments performed on translations generated by all four MT systems which have been available. Section 7.2.3 contains results achieved on translation hypotheses from the alignment templates system, the phrase-based translation system, and by Systran. Here, the complete set of experiments including comparative results on all language pairs from the TT2 Xerox collection will be shown. Additionally, confidence estimation for finite state transducer translations has been performed on the German-English language pair.

As stated in section 7.1, hypotheses from four different machine translation systems have been used as input for the confidence estimation. Those hypotheses have been generated by

- the phrase-based translation system described in section 3.3.3.1,
- the alignment templates system described in appendix C.1,
- the finite state transducer system (see section C.2),
- the Systran version available on the Altavista web-page in June 2005 [sys 05].

These experiments have been performed to see whether the direct phrase-based confidence measures discriminate well on output from different translation systems. Especially the translations from Systran will yield insight on how much the classification performance depends on the underlying machine translation system.

The evaluation of the system-independent confidence measures for all four different translation systems is shown in table D.1. The translation direction is German → English.

The investigated methods are the ones which consider only the single best translation and no additional system output: based on IBM model 1, HMMs and the direct phrase-based method. All measures distinctly decrease the CER compared to the baseline, with significance at the 1%-level. For the three statistical machine translation systems, the direct phrase-based measures outperform the IBM model 1 and HMM. This tendency is clear both in terms of CER and IROC. The largest gain in terms of CER is achieved for the translations generated by the alignment templates and the PBT system: The reduction is almost 16% relative over the best other measure. Obviously, the direct phrase-based confidence measures perform best for phrase-based translation systems. The picture looks different for the translations generated by Systran: All three confidence measures discriminate similarly well. In terms of CER, the IBM model 1 is slightly better, whereas the HMM achieves the highest IROC value.

Table D.1: Classification performance in terms of CER[%] and IROC[%] for different system independent confidence measures on the TT2 Xerox German → English test set, version: raw. Reference based on WER. Hypotheses from different MT systems.

model	AT		PBT		FST		Systran	
	CER	IROC	CER	IROC	CER	IROC	CER	IROC
baseline	49.2	-	48.4	-	46.6	-	37.4	-
99%-confidence interval	±2.2	-	±2.4	-	±2.4	-	±1.4	-
IBM-1 (max)	32.7	73.3	32.8	72.2	34.6	70.3	**23.6**	80.7
mon. HMM & bigram	34.0	71.8	31.3	74.9	37.1	67.3	23.8	**81.5**
direct phrase-based	**27.6**	**79.1**	**26.4**	**80.3**	**30.2**	**75.0**	24.3	81.4

More comparative experiments have been carried out on the other two language pairs from the TransType2 Xerox collection: French-English and Spanish-English. The results are shown in table D.2. The tendencies are the same as for German → English: The direct phrase-based confidence measures outperform the simpler models. They improve classification significantly (at the 1%-level) over the baseline. The IBM model 1 and HMM show worse discriminative power, especially on Spanish → English. On this language pair, the inverted HMM instead of the monotone one has been applied because it performs slightly better. A comparison of CER and IROC for the IBM model 1 and HMM shows discrepancies: The IBM model 1 yields about the same CER as the HMM for the statistical machine translation systems, but the IROC is much higher. Obviously, the IBM model 1 could discriminate better. But the acceptance threshold estimated on the development data differs largely from the one which is optimal for the test set, resulting in high CER.

D.2 Data analysis

Two different data analyses have been performed in this thesis. The first investigates how well the phrase models learned in training cover the development and test data on two different translation tasks. This study is carried out to explore the relation between

Table D.2: Classification performance in terms of CER[%] and IROC[%] for different direct confidence measures on the TT2 Xerox French → English and Spanish → English test sets. Reference based on WER. Hypotheses from different MT systems.

task	model	AT		PBT		Systran	
		CER	IROC	CER	IROC	CER	IROC
F → E	baseline	42.5	-	42.2	-	32.8	-
	99%-confidence interval	±2.3	-	±2.3	-	±1.7	-
	IBM-1 (max)	34.1	68.3	35.6	66.9	26.0	81.3
	monotone HMM & bigram	31.3	**73.2**	33.2	70.8	31.9	55.9
	direct phrase-based	**30.2**	73.0	**30.6**	**74.4**	**22.7**	**83.2**
S → E	baseline	20.8	-	19.2	-	43.7	-
	99%-confidence interval	±1.9	-	±2.0	-	±1.5	-
	90%-confidence interval	±1.2	-	±1.3	-	±1.0	-
	IBM-1 (max)	20.0	66.8	18.3	73.2	21.7	85.5
	inverted HMM & bigram	20.3	60.8	18.2	69.0	32.2	81.8
	direct phrase-based	**17.5**	**76.0**	**16.4**	**77.0**	**17.3**	**87.5**

this coverage and the quality of the direct phrase-based confidence measure. The second analysis examines the distribution of sentence lengths on the development and test sets for four different corpora. This is relevant for the definition of separate acceptance thresholds based on the sentence length as discussed in section 4.5. The investigation will show whether any improvement of the classifier can be expected from this method.

D.2.1 Phrase lexica

The goal of the data analysis presented in the following is to investigate why the direct phrase-based confidence measures outperform all other confidence measures on the TT2 Xerox corpora. The Spanish-English data from this task and from the EPPS database has been studied. The analysis assesses how well the development and test data are captured by the phrases extracted from the training corpus. To this end, those sentence pairs are counted where the source sentence and the translation generated by the phrase-based translation system are completely covered by a bilingual phrase.

Table D.3 shows these numbers for the development and test sets of the two translation tasks. One can see that a much larger portion of the TT2 corpora matches the training data: 29.5% and 6.7% of the words in the development and test hypotheses, respectively, are generated from a perfect match with a bilingual phrase. These sentences are relatively long, especially on the development corpus, with an average of 9.95 words. Here, the phrase lexicon actually serves as a translation memory storing the training data. Note that there are also examples in the data where the source sentence occurs in a bilingual phrase, but with a different translation. In these cases, the phrase pair has a relatively low probability so that the system generated a hypothesis by combining several shorter

phrases instead.

The analysis of the EPPS data shows that there exist by far fewer sentence pairs which are completely covered by the training data: 7.3% and 1.6% percent of the words on development and test corpus, respectively. Moreover, these matching sentences are very short, with less than 3 words on average.

The last column of table D.3 shows the CER on the two types of sentence pairs. This indicates how many of the words are incorrect when compared to the reference translations using m-WER. As to be expected, the error rate on the covered sentence pairs is very low on the TT2 data, with only 0.8% on the development corpus. Interestingly, this is completely different on the EPPS corpora: 18.2% of the words are incorrect on the development set. A reason for this might be that the translations of the development and test corpora have been produced by different translators than those of the training corpus: The training corpus consists of data translated by the official translators of the European Parliament, whereas the other two sets have been generated by translators hired by the project partner ELDA (Evaluations and Language resources Distribution Agency). They probably generated translations which differ notably in style. In addition to that, the sentences or phrases might occur in different contexts requiring different translations. The TT2 documents on the other hand come from a restricted domain and from the same collection of data. Therefore, they are more coherent w.r.t. style.

The fact that a relatively large portion of the TT2 data are covered by one bilingual phrase makes it easier to understand why the direct phrase-based confidence measures discriminate so well: There are many cases in which no or few context outside the phrase is needed. Moreover, the phrases represent the data well so that they can be used as reliable indicators of correctness.

Table D.3: Data analysis: number of sentence pairs in the development and test corpus contained in the phrase lexicon, and percentage of incorrect words in these sets according to m-WER. TT2 Xerox Spanish → English, version: simplified; EPPS Spanish → English, version: verbatim.

corpus		sentence pair in phrase lexicon?	# sentences	# target words	avg. sent. length	CER [%]
TT2	Dev	yes	423	4 208	9.95	0.8
		no	568	10 047	17.06	18.0
	Test	yes	138	568	4.12	7.6
		no	987	7 932	8.04	20.1
EPPS	Dev	yes	595	1 448	2.43	18.2
		no	2 048	18 471	9.02	27.7
	Test	yes	92	273	2.97	7.3
		no	981	17 327	17.67	30.6

D.2.2 Sentence length distributions

To see how much gain can expected from classifiers which take the source sentence or target hypothesis length into account (see section 4.5), a data analysis has been performed. The thresholds are optimized on the development set and applied on the test set. Thus, a similar distribution of the sentence lengths on the two sets is required to achieve reliable estimates of the thresholds.

Figure D.1 shows the distribution of source sentence lengths on the three investigated TransType2 corpora and on the EPPS Spanish data. The vertical lines mark the average source sentence length for each of the corpora. It can be clearly seen that on the French and the German TT2 corpora, the frequencies of the different source sentence lengths are relatively similar on the development and the test sets. For French, the average source sentence lengths are even identical on the two sets. For the two Spanish corpora on the other hand, the distributions differ significantly between development and test corpora: On the TT2 Xerox corpus, the test sentences are much shorter on average than those in the development corpus. In the EPPS development corpus, the source sentences are not longer than 30 words, whereas the test sentences are up to 60 words long. Moreover, the lengths are much more equally distributed in the test than in the development data. This large discrepancy will make it hard to obtain a reliable estimate of the separate thresholds, even with smoothing. Thus, it cannot be expected to achieve a significant improvement from determining separate classification thresholds for each source sentence length on the two Spanish-English corpora. Moreover, on the EPPS data there are no examples of long sentences in the development data, so that the estimation of the threshold for these sentences will be very unreliable. An analysis of the automatic translations on these corpora has shown the same effects for the hypothesis lengths.

As can be seen in figure D.1, the largest gain (if any) through the use of sentence length based classifiers is to be expected on the French → English and German → English translation tasks. Therefore, the detailed experiments presented in section 7.2.7 have been performed on the TT2 Xerox French → English data.

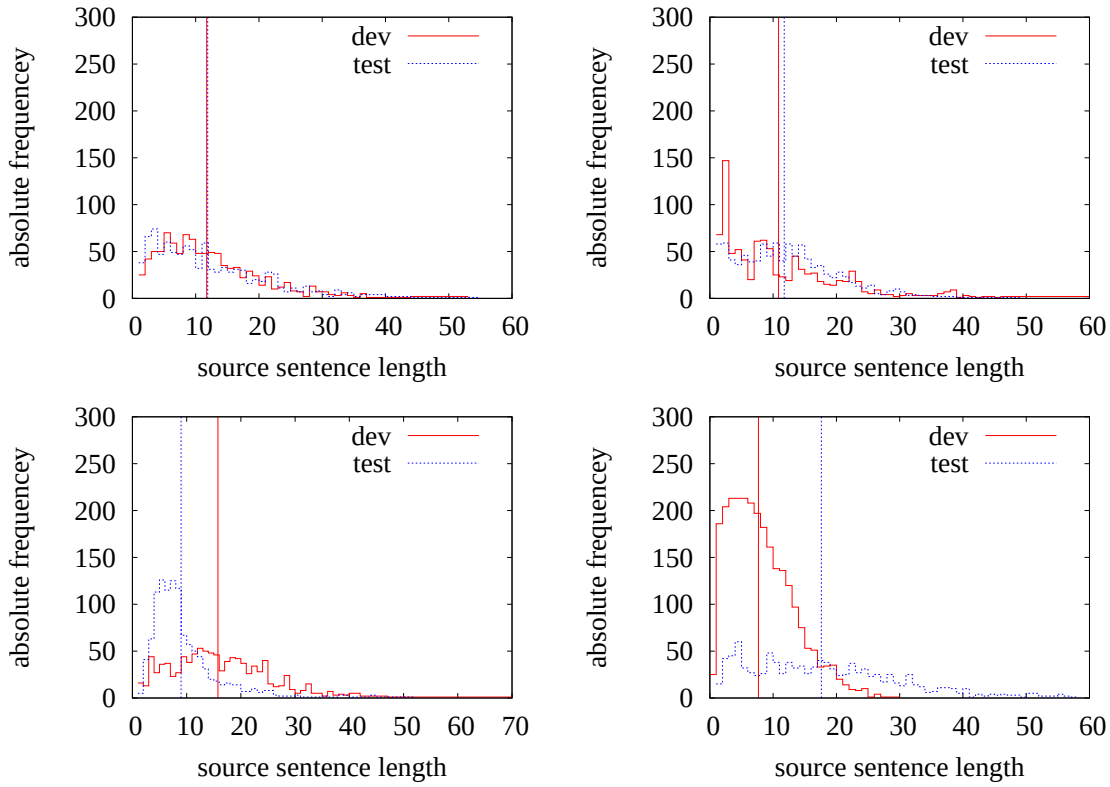


Figure D.1: Distribution of source sentence lengths on development and test corpora. Vertical lines mark the average lengths. **Top left:** TT2 Xerox French (raw), **top right:** TT2 Xerox German (raw), **bottom left:** TT2 Xerox Spanish (simplified), **bottom right:** EPPS Spanish (verbatim).

E Symbols and acronyms

E.1 Mathematical symbols

f_1^J	source sentence
e_1^I	target sentence
$e_{n1}^{nI_n}$	target sentence at rank n in an N-best list
e_1^i	prefix of the target sentence
f_j	source word in position j of the sentence
e_i	target word in position i of the sentence
a_1^J	word alignment mapping source to target positions
B_1^I	word alignment mapping target positions to sets of source positions
n_e	count of word e in the sentence e_1^I
$\tilde{n}_e$	count of word e in the sentence $\tilde{e}_1^{\tilde{I}}$
$\mathcal{L}(e_1^I, \tilde{e}_1^{\tilde{I}})$	Levenshtein alignment between sentences e_1^I and $\tilde{e}_1^{\tilde{I}}$
$\mathcal{L}_i(e_1^I, \tilde{e}_1^{\tilde{I}})$	Levenshtein alignment of word e_i (between sentences e_1^I and $\tilde{e}_1^{\tilde{I}}$)
$d(e_1^I, \tilde{e}_1^{\tilde{I}})$	number of deletions in the Levenshtein alignment between sentences e_1^I and $\tilde{e}_1^{\tilde{I}}$
n', n	nodes of a word graph
(n', n)	edge in a word graph
n_0	root node of a word graph
$\Phi(\dots)$	forward probability
$\Psi(\dots)$	backward probability
$Q(\dots)$	auxiliary quantity
$\delta(\cdot, \cdot)$	Kronecker delta
$\delta(\cdot)$	extension of Kronecker delta: $\delta(a) = \begin{cases} 1 & \text{if a is true} \\ 0 & \text{otherwise} \end{cases}$
F_p	prediction F-measure

E.2 Acronyms

AT	**a**lignment **t**emplate
BLEU	**bil**ingual **e**valuation **u**nderstudy
CA	**c**orrect **a**cceptance
CER	**c**lassification **e**rror **r**ate
CLSP	**C**enter for **L**anguage and **S**peech **P**rocessing

CR	**c**orrect **r**ejection
EPPS	**E**uropean **P**arliament **P**lenary **S**essions
EU	**E**uropean **U**nion
FSA	**f**inite **s**tate **a**utomaton
FST	**f**inite **s**tate **t**ransducer
HMM	**h**idden **M**arkov **m**odel
IROC	**i**ntegral of the **r**eceiver **o**perating **c**haracteristic curve
ISI	**I**nformation **S**ciences **I**nstitute at University of Southern California
IWSLT	**I**nternational **W**orkshop on **S**poken **L**anguage **T**ranslation
KSR	**k**ey**s**troke **r**atio
LM	**l**anguage **m**odel
MLP	**m**ulti-**l**ayer **p**erceptron
MT	**m**achine **t**ranslation
m-WER	**m**ulti-reference **w**ord **e**rror **r**ate
m-PER	**m**ulti-reference **p**osition independent word **e**rror **r**ate
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
NLP	**n**atural **l**anguage **p**rocessing
OOV	**o**ut **o**f **v**ocabulary
PBT	**p**hrase **b**ased **t**ranslation
PER	**p**osition independent word **e**rror **r**ate
ROC	**r**eceiver **o**perating **c**haracteristic
SMT	**s**tatistical **m**achine **t**ranslation
TT2	**T**rans**T**ype**2**
TC-STAR	**T**echnology and **C**orpora for **S**peech to **S**peech **T**ranslation
WER	**w**ord **e**rror **r**ate

Bibliography

[Akiba et al. 04a] Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, J. Tsujii: Overview of the IWSLT04 Evaluation Campaign. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 1–12, Kyoto, Japan, September/October 2004.

[Akiba et al. 04b] Y. Akiba, E. Sumita, H. Nakaiwa, S. Yamamoto, H.G. Okuno: Using a Mixture of N-Best Lists from Multiple MT Systems in Rank-Sum-Based Confidence Measure for MT Outputs. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pp. 322–328, Geneva, Switzerland, August 2004.

[Alshawi 96] H. Alshawi: Head Automata and Bilingual Tiling: Translation with Minimal Representations. In *Proc. 34th Annual Meeting of the Assoc. for Computational Linguistics*, pp. 167 – 176, Santa Cruz, CA, June 1996.

[Bangalore and Riccardi 00] S. Bangalore, G. Riccardi: Stochastic Finite-State models for Spoken Language Machine Translation. In *Proc. Workshop on Embedded Machine Translation Systems, NAACL*, pp. 52–59, Seattle, WA, May 2000.

[Bender et al. 04] O. Bender, R. Zens, E. Matusov, H. Ney: Alignment Templates: the RWTH SMT System. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 79–84, Kyoto, Japan, September 2004.

[Bertoldi et al. 04] N. Bertoldi, R. Cattoni, M. Cettolo, M. Federico: The ITC-irst Statistical Machine Translation System for IWSLT-2004. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 51–58, Kyoto, Japan, September 2004.

[Bisani and Ney 04] M. Bisani, H. Ney: Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 409–412, Montreal, Canada, May 2004.

[Blatz et al. 03] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, N. Ueffing: Confidence Estimation for Machine Translation. Final report, JHU/CLSP Summer Workshop, 2003. `http://www.clsp.jhu.edu/ws2003/groups/estimate/`.

[Blatz et al. 04] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, N. Ueffing: Confidence Estimation for Machine Translation. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pp. 315–321, Geneva, Switzerland, August 2004.

[Brown et al. 93] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, R.L. Mercer: The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, Vol. 19, No. 2, pp. 263–311, June 1993.

[Casacuberta et al. 01] F. Casacuberta, D. Llorens, C. Martínez, S. Molau, F. Nevado, H. Ney, M. Pastor, D. Picó, A. Sanchis, E. Vidal, J.M. Vilar: Speech-to-speech translation based on finite-state transducers. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 613–616, Salt Lake City, UH, May 2001.

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

[Civera et al. 04] J. Civera, J.M. Vilar, E. Cubel, A.L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, J. González: From machine translation to computer assisted translation using finite-state models. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 349–356, Barcelona, Spain, July 2004.

[Collobert et al. 02] R. Collobert, S. Bengio, J. Mariéthoz.: Torch: a modular machine learning software library. Technical Report IDIAP-RR 02-46, IDIAP, 2002.

[Ding and Palmer 05] Y. Ding, M. Palmer: Machine Translation Using Probabilistic Synchronous Dependency Insertion Grammars. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 541–548, Ann Arbor, MI, June 2005.

[Doddington 02] G. Doddington: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. ARPA Workshop on Human Language Technology*, pp. 128–132, San Diego, 2002.

[Duda et al. 01] R. Duda, P. Hart, D. Stork: *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 2001.

[Foster et al. 02] G. Foster, P. Langlais, G. Lapalme: TransType: Text Prediction for Translators. In *Proceedings of the ACL-02 Demonstrations Session*, pp. 93–94, Philadelphia, PA, July 2002.

[Gandrabur and Foster 03] S. Gandrabur, G. Foster: Confidence Estimation for Text Prediction. In *Proc. Conf. on Natural Language Learning (CoNLL)*, pp. 95–102, Edmonton, Canada, May 2003.

[Goel and Byrne 03] V. Goel, W. Byrne: Minimum Bayes-Risk Automatic Speech Recognition. In W. Chou, B.H. Juang, editors, *Pattern Recognition in Speech and Language Processing*. CRC Press, 2003.

[Jayaraman and Lavie 05] S. Jayaraman, A. Lavie: Multi-Engine Machine Translation Guided by Explicit Word Matching. In *Proc. of the 10th Annual Conf. of the European Association for Machine Translation (EAMT)*, pp. 143–152, Budapest, Hungary, May 2005.

[Kanthak and 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.

[Kanthak et al. 05] S. Kanthak, D. Vilar, E. Matusov, R. Zens, H. Ney: Novel Reordering Approaches in Phrase-Based Statistical Machine Translation. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pp. 167–174, Ann Arbor, MI, June 2005.

[Koehn et al. 03] P. Koehn, F.J. Och, D. Marcu: Statistical Phrase-Based Translation. In *Proc. of the Human Language Technology Conf. (HLT-NAACL)*, pp. 127–133, Edmonton, Canada, May/June 2003.

[Kumar and Byrne 03] S. Kumar, W. Byrne: A weighted finite state transducer implementation of the alignment template model for statistical machine translation. In *Proc. of the Human Language Technology Conf. (HLT-NAACL)*, pp. 142–149, Edmonton, Canada, May/June 2003.

[Kumar and Byrne 04] S. Kumar, W. Byrne: Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proc. of the Human Language Technology Conf. (HLT-NAACL)*, pp. 169–176, Boston, MA, May 2004.

[Leusch et al. 03] G. Leusch, N. Ueffing, H. Ney: A Novel String-to-String Distance Measure with Applications to Machine Translation Evaluation. In *Proc. MT Summit IX*, pp. 240–247, New Orleans, LA, September 2003.

[Leusch et al. 05] G. Leusch, N. Ueffing, D. Vilar, H. Ney: Preprocessing and Normalization for Automatic Evaluation of Machine Translation. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pp. 17–24, Ann Arbor, MI, June 2005. Association for Computational Linguistics.

[Levenshtein 66] V.I. Levenshtein: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, Vol. 10, No. 8, pp. 707–710, Feb. 1966.

[Lin 04] D. Lin: A Path-based Transfer Model for Machine Translation. In *COLING '04: The 20th Int. Conf. on Computational Linguistics*, pp. 625–630, Geneva, Switzerland, Aug 2004.

[Mariño et al. 05] J.B. Mariño, R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A. Fonollosa, M. Ruiz: Bilingual N-gram Statistical Machine Translation. In *Proc. MT Summit X*, pp. 275–282, Phuket, Thailand, September 2005. Association for Computational Linguistics.

[Mathar 96] R. Mathar: *Informationstheorie*. Aachener Beiträge zur Mathematik. Verlag der Augustinus Buchhandlung, Aachen, 1996.

[Menezes and Richardson 01] A. Menezes, S.D. Richardson: A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora. In *39th Annual Meeting of the Assoc. for Computational Linguistics - joint with EACL 2001: Proc. Workshop on Data-Driven Machine Translation*, pp. 39–46, Toulouse, France, July 2001.

[Ney et al. 04] H. Ney, M. Popović, D. Sündermann: Error Measures and Bayes Decision Rules Revisited with Applications to POS Tagging. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 270–276, Barcelona, Spain, July 2004.

[Ney et al. 05] H. Ney, V. Steinbiss, R. Zens, E. Matusov, J. Gonzalez, Y.S. Lee, S. Roukos, M. Federico, M. Kolss, R. Banchs: TC-STAR Deliverable no. D5: SLT Progress Report. Technical report, Integrated project TC-STAR (IST-2002-FP6-506738) funded by the European Commission, May 2005. `http://www.tc-star.org/`.

[Nießen et al. 00] S. Nießen, F.J. Och, G. Leusch, H. Ney: An evaluation tool for machine translation: Fast evaluation for MT research. In *Proc. of the Second Int. Conf. on Language Resources and Evaluation (LREC)*, pp. 39–45, Athens, Greece, May 2000.

[NIST 00] NIST: The 2000 NIST Evaluation Plan for Recognition of Conversational Speech over the Telephone. `http://www.nist.gov/speech/tests/ctr/h5_2000/h5-2000-v1.3.htm`, 2000.

[NIST 02] NIST: Machine Translation Evaluation Chinese–English. `http://nist.gov/speech/tests/mt/`, 2002.

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

[Och and 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 and Ney 04] F.J. Och, H. Ney: The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, Vol. 30, No. 4, pp. 417–449, December 2004.

[Och et al. 03] F.J. Och, R. Zens, H. Ney: Efficient Search for Interactive Statistical Machine Translation. In *Proc. 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pp. 387–393, Budapest, Hungary, April 2003.

[Papineni et al. 02] K. Papineni, S. Roukos, T. Ward, W.J. Zhu: BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, Philadelphia, PA, July 2002.

[Press et al. 02] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery: *Numerical Recipes in C++*. Cambridge University Press, Cambridge, UK, 2002.

[Quirk 04] C. Quirk: Training a Sentence-Level Machine Translation Confidence Metric. In *Proc. of the Fourth Int. Conf. on Language Resources and Evaluation (LREC)*, pp. 825–828, Lisbon, Portugal, May 2004.

[Sanchis 04] A. Sanchis: *Estimación y aplicación de medidas de confianza en reconocimiento automático del habla*. Ph.D. thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, May 2004.

[Schoenemann 05] T. Schoenemann: *Model-based Confidence Measures for Statistical Machine Translation*. Diploma thesis, RWTH Aachen University, Aachen, Germany, June 2005.

[Stolcke 02] A. Stolcke: SRILM – An Extensible Language Modeling Toolkit. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, Vol. 2, pp. 901–904, Denver, CO, 2002.

[Sumita et al. 04] E. Sumita, Y. Akiba, T. Doi, A. Finch, K. Imamura, H. Okuma, M. Paul, M. Shimohata, T. Watanabe: EBMT, SMT, Hybrid and More: ATR Spoken Language Translation System. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 13–20, Kyoto, Japan, September 2004.

[sys 05] Systran MT system, June 2005. `http://babelfish.altavista.com/tr`.

[tcs 05] TC-STAR - Technology and Corpora for Speech to Speech Translation, 2005. Integrated project TC-STAR (IST-2002-FP6-506738) funded by the European Commission. `http://www.tc-star.org/`.

[Tillmann et al. 97] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, H. Sawaf: Accelerated DP Based Search for Statistical Translation. In *European Conf. on Speech Communication and Technology*, pp. 2667–2670, Rhodes, Greece, September 1997.

[tt2 05] TransType2 - Computer Assisted Translation, 2005. RTD project TransType2 (IST–2001–32091) funded by the European Commission. `http://tt2.atosorigin.es/`.

[Ueffing and Ney 03] N. Ueffing, H. Ney: Using POS Information for Statistical Machine Translation into Morphologically Rich Languages. In *Proc. 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pp. 347–354, Budapest, Hungary, April 2003.

[Ueffing and Ney 04] N. Ueffing, H. Ney: Bayes Decision Rule and Confidence Measures for Statistical Machine Translation. In *Proc. EsTAL - España for Natural Language Processing*, pp. 70–81, Alicante, Spain, October 2004. Lecture Notes in Computer Science, Springer Verlag.

[Ueffing and Ney 05a] N. Ueffing, H. Ney: Application of Word-Level Confidence Measures in Interactive Statistical Machine Translation. In *Proc. of the 10th Annual Conf. of the European Association for Machine Translation (EAMT)*, pp. 262–270, Budapest, Hungary, May 2005.

[Ueffing and Ney 05b] N. Ueffing, H. Ney: Word-Level Confidence Estimation for Machine Translation using Phrase-Based Translation Models. In *Proc. of the Human Language Technology Conf./Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp. 763–770, Vancouver, Canada, October 2005.

[Ueffing et al. 02] N. Ueffing, F.J. Och, H. Ney: Generation of Word Graphs in Statistical Machine Translation. In *Proc. of the Conf. on Empirical Methods for Natural Language Processing (EMNLP)*, pp. 156–163, Philadelphia, PA, July 2002.

[Ueffing et al. 03] N. Ueffing, K. Macherey, H. Ney: Confidence Measures for Statistical Machine Translation. In *Proc. MT Summit IX*, pp. 394–401, New Orleans, LA, September 2003.

[Vidal 97] E. Vidal: Finite-State Speech-to-Speech Translation. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 111–114, Munich, Germany, April 1997.

[Vilar et al. 05] D. Vilar, E. Matusov, S. Hasan, R. Zens, H. Ney: Statistical Machine Translation of European Parliamentary Speeches. In *Proc. MT Summit X*, pp. 259–266, Phuket, Thailand, September 2005.

[Vogel et al. 96] S. Vogel, H. Ney, C. Tillmann: HMM-Based Word Alignment in Statistical Translation. In *COLING '96: The 16th Int. Conf. on Computational Linguistics*, pp. 836–841, Copenhagen, Denmark, August 1996.

[Vogel et al. 04] S. Vogel, S. Hewavitharana, M. Kolss, A. Waibel: The ISL Statistical Translation System for Spoken Language Translation. In *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, pp. 65–72, Kyoto, Japan, September 2004.

[Wahlster 00] W. Wahlster, editor: *Verbmobil: Foundations of speech-to-speech translations*. Springer Verlag, Berlin, Germany, July 2000.

[Weintraub et al. 97] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, A. Stolcke: Neural - Network Based Measures of Confidence for Word Recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 887 – 890, Munich, Germany, April 1997.

[Wessel 02] F. Wessel: *Word Posterior Probabilities for Large Vocabulary Continuous Speech Recognition*. Ph.D. thesis, RWTH Aachen University, Aachen, Germany, January 2002.

[Wessel et al. 01] F. Wessel, R. Schlüter, K. Macherey, H. Ney: Confidence Measures for Large Vocabulary Continuous Speech Recognition. *IEEE Trans. on Speech and Audio Processing*, Vol. 9, No. 3, pp. 288–298, March 2001.

[Yamada and Knight 02] K. Yamada, K. Knight: A Decoder for Syntax-based Statistical MT. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 303–310, Philadelphia, PA, July 2002.

[Zens and Ney 04] R. Zens, H. Ney: Improvements in Phrase-Based Statistical Machine Translation. In *Proc. of the Human Language Technology Conf. (HLT-NAACL)*, pp. 257–264, Boston, MA, May 2004.

[Zens and Ney 05] R. Zens, H. Ney: Word Graphs for Statistical Machine Translation. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pp. 191–198, Ann Arbor, MI, June 2005.

[Zens et al. 02] R. Zens, F.J. Och, H. Ney: Phrase-Based Statistical Machine Translation. In M. Jarke, J. Koehler, G. Lakemeyer, editors, *25th German Conf. on Artificial Intelligence (KI2002)*, Vol. 2479 of *Lecture Notes in Artificial Intelligence (LNAI)*, pp. 18–32, Aachen, Germany, September 2002. Springer Verlag.

Lebenslauf – Curriculum Vitae

Angaben zur Person

Name	Nicola Ueffing
Anschrift	Stromgasse 28-32, 52064 Aachen
E-Mail	Nicola.Ueffing@web.de
Geburtsdatum	20. Juli 1975
Geburtsort	Aachen, Deutschland
Staatsangehörigkeit	deutsch

Schulbildung

1981 – 1995	Grundschule Eschweiler-Bergrath
1985 – 1994	Städtisches Gymnasium Eschweiler (Abitur)

Studium

Oktober 1994 – August 2000	Mathematikstudium an der RWTH Aachen Vertiefungsfach: Graphentheorie Abschluss als Diplom-Mathematikerin
September 1996 – März 1997	Mathematikstudium am University College Dublin, Irland
September 2000 – März 2006	Promotionsstudium an der RWTH Aachen
Juni – August 2003	Teilnahme am Summer Workshop des Centre for Language and Speech Processing an der Johns Hopkins University, Baltimore, MD

Arbeitstätigkeiten

1997 – 2000	Studentische Hilfskraft am Institut für Statistik und Wirtschaftsmathematik, RWTH Aachen
September 2000 - Februar 2006	Wissenschaftliche Angestellte am Lehrstuhl für Informatik VI (Sprachverarbeitung und Mustererkennung), RWTH Aachen
seit April 2006	Research Associate beim National Research Council Canada, Gatineau, Québec