Developing Deployable Spoken Language Translation Systems given Limited Resources

zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
von der Fakultät für Informatik
der Universität Fridericiana zu Karlsruhe (TH)

genehmigte

Dissertation

von
Matthias Eck
aus Miltenberg

Tag der mündlichen Prüfung: 7. November 2008

Erster Gutachter: Prof. Dr. Alexander Waibel
Zweiter Gutachter: Prof. Dr. Tanja Schultz

This dissertation is dedicated to my parents.

Acknowledgements

Many people have helped me in my progress towards this thesis. First of all, it is my advisor Alex Waibel, who introduced me into the field of statistical machine translation. His patience, trust and challenges have helped and motivated me in my research, and he inspired me with his vision.

Many thanks to my supervisor and mentor Stephan Vogel for so many in-depth discussions over all the years. I learned so much from his tremendous experience in machine translation and his unique ability to share it.

Special thanks to co-advisor Tanja Schultz for a fresh perspective. Her insightful questions and detailed comments on my dissertation helped improve the quality of this thesis work.

Bing Zhao has been has been a great colleague over many years and a fun office mate.

Freya Fridy deserves a special mention for careful proofreading and greatly improving the dissertation and its punctuation.

I am very grateful to everyone else at LTI/InterACT at the University of Karlsruhe and at Carnegie Mellon University, in particular Nguyen Bach, Norbert Berger, Michael Bett, Alan Black, Susanne Burger, Celine Carraux, Paisarn Charoenpornsawat, Silke Dannenmaier, Anthony D'Auria, Christian Fügen, Qin Gao, Nimish Gautam, Petra Gieselmann, Linda Hager, Sven Haidan, Sanjika Hewavitharana, Silja Hildebrand, Hartwig Holzapfel, Chiori Hori, Roger Hsiao, Fei Huang, Qin Jin, Szu-Chen (Stan) Jou, Thilo Köhler, Muntsin Kolss, Florian Kraft, Lisa Krieg, Ian Lane, Kornel Laskowski, Alon Lavie, Victoria MacLaren, Robert Malkin, Lisa Mauti, Kristen Messinger, Florian Metze, Thuylinh Nguyen, Kai Nickel, Jan Niehues, Mohamed Noamany, Yue Pan, Matthias Paulik, Kay Peterson, Martin Raab, Radha Rao, Narges Razavian, Jürgen Reichert, Margit Rödder, Ivica Rogina, Annette Römer, Kay Rottmann, Thomas Schaaf, Sebastian Stüker, Rainer Stiefelhagen, Yik-Cheung (Wilson) Tam, Alicia Tribble, Stephen Valent, Dan Valsan, Ashish Venugopal, Matthias Wölfel, Hua Yu, Ying (Joy) Zhang, Andreas Zollmann.

It has been a pleasure to work with all of you and I look forward to working with you in the future.

I owe a lot to my friends and family for their continued support.

Zusammenfassung

Die Leistung maschineller Übersetzungssysteme in Forschungseinrichtungen hat in den letzten Jahren beträchtliche Fortschritte gemacht. Dies ist insbesondere für Statistische Maschinenübersetzung der Fall. Seit ihrer Einführung, beginnend etwa 1990, erreichte diese Technik meist bessere Ergebnisse als alle anderen Verfahren und liefert derzeit beachtliche Übersetzungsleistungen.

Insbesondere in eingeschränkten Themenbereichen, wie touristischen Dialogen und medizinischen und militärischen Anwendungen, könnten die Übersetzungen bereits in realen Situationen sehr nützlich sein. Man ist hier nicht mehr nur auf Forschungseinrichtungen und Vorführungen beschränkt.

Die Frage ist nicht mehr länger nur, wie die Technologie auf einen angemessenen Level gebracht werden kann, sondern auch, wie die existierende Technik tatsächlich eingesetzt werden kann. Wie können Systeme entwickelt werden, die Touristen, Mediziner und Soldaten tatsächlich sinnvoll benutzen können und wollen?

Die folgenden Punkte sind unserer Ansicht nach die wichtigsten Unterschiede zwischen einem bestehenden Forschungssystem und einem System, das tatsächlich eingesetzt werden könnte:

	Forschungssystem	Einsetzbares System
Sprachen	Wenige Sprachenpaare	Viele Sprachenpaare
Hardware	High-End Server	Notebook, Mobile Geräte
Anwendung	Evaluationen, Demonstrationen	Kommunikation
Evaluation	BLEU/NIST Scores	Kommunikationserfolg
Benutzer	Experten, Forscher	Touristen, Medizin, Militär
Schnittstelle	Komplex, Kommandozeile	Einfach, Interaktiv

Ein wichtiger Unterschied zwischen einem Forschungssystem und einem einsetzbaren System ist die Anzahl der unterstützten Sprachen. Die meisten Forschungseinrichtungen konzentrieren sich auf wenige Sprachenpaare, für die größere Korpora verfügbar sind. Systeme, die für Touristen und vor allem für medizinische Katastrophenhilfe oder militärische Zwecke verwendet

werden sollen, müssten eine viel größere Anzahl von Sprachen unterstützen. Insbesondere ist es notwendig, schnell Systeme für neue Sprachen zu entwickeln, die in den Mittelpunkt des Interesses rücken, z.B. aufgrund von medizinischen oder militärischen Einsätzen.

Übersetzungssysteme in Forschungseinrichtungen laufen in der Regel auf leistungsfähigen Servern, um größere und komplexere Modelle nutzen zu können. Auch kleine Leistungsverbesserungen sollen realisiert werden. Die Benutzer sind Forscher und Experten, deren Ziel nicht die Einfachheit der Benutzeroberfläche ist. Ziel ist Flexibilität, die Möglichkeit einer einfachen Integration von zusätzlichen Komponenten und das schnelle Durchführen von Experimenten.

Für Benutzer außerhalb der Forschungseinrichtungen in den genannten Anwendungsgebieten Tourismus, Medizin und Militär ist es unerlässlich, dass das Gerät tatsächlich tragbar ist. Ein Notebook-Computer könnte in einigen Situationen akzeptabel sein, aber das Ziel wäre in der Regel die Übersetzung auf einem PDA, einer tragbaren Spielkonsole oder auf einem Mobiltelefon. Touristen, medizinisches und militärisches Personal werden auch keine Benutzeroberfläche tolerieren, die ihre kommunikativen Fähigkeiten einschränkt und ein einfach zu bedienendes System fordern. Allerdings sind einfache Benutzerschnittstellen verfügbar und wurden bereits mehrfach demonstriert, weshalb sie hier nicht weiter berücksichtigt werden.

Übersetzungssysteme in Forschungseinrichtungen werden in der Regel in Evaluationen oder in Demonstrationen eingesetzt. Die Testsätze in diesen Situationen sind entweder zur Veranschaulichung einer spezifischen Fähigkeit ausgewählt oder sind standardisiert, um Vergleiche zwischen verschiedenen Gruppen anzustellen. Die Testsätze versuchen natürlich die tatsächliche Aufgabe zu emulieren, verfehlen dieses Ziel aber oft, insbesondere im Hinblick auf Eigennamen und Spezialvokabular. Dies wird durch die Standard-Metriken unterstützt, die nur wenig Wert auf die korrekte Übersetzung eines individuellen Namens legen. Andererseits ist es für einen tatsächlichen Benutzer von entscheidender Bedeutung, Eigennamen übersetzen zu können, um erfolgreich zu kommunizieren. Eigennamen sind oft die wichtigsten Träger von Informationen, und für spezifische Anwendungen wird unter Umständen Spezialvokabular benötigt.
Die drei wichtigsten Themen dieser Arbeit sind nach dieser Analyse:

- Portabilität für die schnelle und kostengünstige Übertragung auf neue Sprachpaare
- Übersetzungsmodelle für kleine, tragbare Geräte
- Verbesserung der Abdeckung von Eigennamen und Spezialvokabular

Portabilität auf neue Sprachenpaare Der wichtigste Kostenfaktor bei der Portierung eines statistischen Übersetzungssystems auf ein neues Sprachenpaar ist die Produktion der Trainingsdaten; hier insbesondere die Übersetzung eines monolingualen Korpus zur Erstellung eines bilingualen Korpus. Zur Begrenzung der Zahl der erforderlichen Übersetzungen wird der Nutzen eines übersetzten Satzes mit verschiedenen Gewichtungstermen abgeschätzt. In formaler Darstellung ist dieses Problem NP-vollständig bzw. NP-hart, aber die spezifische Situation erlaubt die Anwendung eines effizienten Greedy-Algorithmus. Die Gewichtung der Sätze basiert in der Regel auf der Anzahl und Häufigkeit der bisher ungesehenen n-Gramme in dem jeweiligen Satz. Außerdem wird ein informationstheoretischer Ansatz auf Basis von TF-IDF getestet. In einem letzten Ansatz wird der Wert für bereits übersetzte Worte, basierend auf ihren Übersetzungswahrscheinlichkeiten, dynamisch angepasst. Dadurch können mehrdeutige Worte ermittelt werden, für die weitere Sätze übersetzt werden sollten. Nach der Sortierung des monolingualen Korpus gemäss dieser Gewichtungsterme können kleinere Korpora für die menschliche Übersetzung ausgewählt werden. Es kann gezeigt werden, dass Systeme, die auf den kleineren Trainingsdaten trainiert werden, sehr gute Übersetzungsqualität erreichen können. Abhängig von der jeweiligen Situation können die notwendigen Trainingsdaten und damit die Übersetzerkosten um bis zu 75% reduziert werden.

Übersetzungsmodelle für tragbare Geräte Das Hauptproblem, Übersetzungssysteme auf tragbaren Geräten, wie einem PDA, auszuführen, ist der Speicher- und Festplattenbedarf der Übersetzungs- und Sprachmodelle. Die Prozessorgeschwindigkeit selbst ist auch ein Engpass, es konnte aber gezeigt werden, dass Fließkomma-Operationen durch Festpunkt bzw. Integerarithmetik ersetzt werden können, sodass PDA-Prozessoren akzeptable Geschwindigkeiten und nahezu gleiche Übersetzungsqualität erreichen. Das verbleibende Problem ist der Speicherverbrauch der Modelle. Clevere Kodierung und Komprimierungsverfahren können diese erheblich reduzieren. Insbesondere Sprachmodelle können sehr effizient durch eine Bloomfilter-Datenstruktur – eine verlustbehaftete Hashing-Technik – kodiert werden. Die Phrasentabelle kann jedoch nicht so leicht komprimiert werden, und Phrasenpaare müssen entfernt werden, um Speicherplatz einzusparen. Es ist hier das Ziel, genügend Phrasenpaare zu entfernen, um das Modell auf einem tragbaren Gerät zu nutzen und gleichzeitig die bestmögliche Übersetzungqualität zu erreichen. Ein neuer Ansatz zur Entfernung von Phrasenpaaren wird präsentiert, der die Bedeutung eines Phrasenpaares auf Grundlage der tatsächlichen Nutzungsstatistik dieses Phrasenpaares bei der

Übersetzung von großen Mengen von Text abschätzt. Dieser Ansatz wurde durch den Optimal Brain Damage-Algorithmus für neuronale Netze inspiriert. Mehrere Parameter und Konfigurationen werden untersucht, und es kann gezeigt werden, dass die Zahl der Phrasenpaare um bis zu 80% verringert werden kann, ohne wesentliche Auswirkungen auf die Übersetzungsleistung. Dies übertrifft alle bisher bekannten Techniken.

Verbesserung der Abdeckung von Namen und Spezialvokabular Der grundlegende Ansatz, zusätzliche Eigennamen und Spezialvokabular zu sammeln ist die Nutzung anderer Wissensquellen. Im ersten Schritt wird eine medizinische Datenbank verwendet, um die Abdeckung des medizinischen Vokabulars zu verbessern. Hierbei zeigen sich wesentliche Verbesserungen der Übersetzungsqualität. In anderen Bereichen sind diese Datenbanken nicht immer vorhanden oder zugänglich, deshalb ist es hier notwendig, diese Listen – zumindest teilweise – manuell zu generieren.

Es ist nicht effizient, einen großen Wortschatz auf einem tragbaren Gerät mit begrenztem Speicher vorzuhalten, und dies kann unter Umständen die Spracherkennungsleistung eines Sprachübersetzungssystem beeinträchtigen. Aus diesem Grund wird ein Personalisierungsansatz vorgeschlagen. Ein umfangreiches Hintergrundlexikon wird über einen Online-Service bereitgestellt. Die tragbaren Geräte stellen gelegentlich eine Verbindung zu dem Dienst her und aktualisieren ihre Phrasentabellen. Die geschieht auf Grundlage der tatsächlichen Nutzung des jeweiligen Geräts und der aufgetauchten unbekannten Wörter.

Auch mit den zusätzlichen Wortschätzen wird es niemals möglich sein, *jedes* Wort abzudecken. Ein neuer Ansatz wird präsentiert, der monolinguale Lexika und Wörterbücher benutzt, um unbekannte Wörter zu "kommunizieren". Statt des tatsächlich unbekannten Wortes wird dessen Definition übersetzt. Dies führt zu erheblichen Verbesserungen in der Übersetzungsqualität.

Abstract

Automatic machine translation systems in research institutions have reached a considerable level of performance. This is especially true for Statistical Machine Translation. Since its introduction in the 1990s, it has outperformed earlier approaches and produces a translation quality that seemed impossible only a short time ago. Particularly for limited domains like tourism dialogs, medical relief or force protection the translations could already be very useful for real applications outside of laboratories or demonstrations. The question here is no longer how to get the technology to an acceptable level, but how the existing technology can be globally deployed. How can a system be developed that tourists, health professionals or soldiers can actually use and benefit from? The following lists the issues which need to be adressed in order to make a current research system deployable to real users:

	Research system	Deployable system
Languages	Limited number	Many Language Pairs
Hardware	High-End Server	Notebook, Mobile Device
Application	Evaluations, Demonstrations	Actual Communication
Evaluation	BLEU/NIST Scores	Communication success
Users	Expert researchers	Tourists, Medical users, Military users
Interface	Complex, command line	Easy, interactive

The first difference between a research system and a deployable system is the number of generally supported languages. In research most groups focus on a few language pairs for which large training corpora are available. Systems for tourists and especially for medical relief or military uses will have to support a much larger number of languages. It is especially necessary to rapidly support specific language pairs if a sudden demand develops.

In research labs the translation systems are usually run on high-powered, expensive servers to be able to use larger and more complicated models and to realize even small performance improvements. The users are researchers and

experts who do not necessarily care about the simplicity of the user interface as their goal is to be flexible and to allow easy integration of additional components and a fast experimental turnaround. For non-research users especially in the aforementioned instances of tourism, medical and military applications it is essential that the device is actually lightweight and mobile. A notebook computer might be a possibility in some situations but generally the goal would be to have the translation system on a PDA, a handheld game console or even a cell phone. These users, tourists, medical staff and military personnel will also not tolerate a complex user interface affecting their communication abilities and will demand an easy to use system. Simple interfaces are available and various have been demonstrated.

Translation systems in research are usually applied in competitive evaluations or shown as demonstrations. The test sets in these situations are either designed to illustrate a specific capability or are standardized to allow comparisons between different groups. The test sets do indeed try to emulate the actual task but they often fall considerably short of that goal, especially concerning the demand for named entities and specialty vocabulary. This is supported by the standard evaluation metrics that put little weight on an individual named entity. On the other hand, for an actual user being able to translate named entities is crucial for communication success. Named entities are often the main pieces of information in a sentence and specific usages might require specialty vocabulary. After a detailed analysis, the three main topics that will be addressed in this thesis are:

- Low cost portability for fast transfer to new language pairs
- Translation models for small, mobile devices
- Improving named entity and specialty vocabulary coverage

Low Cost Portability The main cost factor when porting a Statistical Machine Translation System to a new language pair is the generation of the training data, especially the translation of a monolingual corpus to produce an aligned bilingual corpus. In order to limit the number of necessary translations, the value of a translated sentence is estimated using various weighting terms. For some formalizations this problem is actually NP complete/NP-hard but the specific situation allows us to apply an efficient greedy algorithm. The weighting schemes for the sentences are generally based on the number and frequency of previously unseen n-grams in the respective sentence. We also use an information theoretic approach basing the value of a sentence on its TF-IDF score. In the last approach we dynamically adjust the value for already seen words on the structure of their IBM Model 1

translation probabilities. This allows us to identify ambiguous words for which additional sentences should be translated to gain more information about their specific word alignment. After sorting the monolingual corpus according to these weighting terms we can select smaller corpora for the human translation and we are able to show that systems trained on much less training data achieve a very competitive performance compared to baseline systems using all available training data. Depending on the actual situation, the necessary training data can be reduced by up to 75%.

Translation Models for Small Devices The main issue in making the translation systems run on a small device like a PDA are the memory requirements of the translation and language models. The processor speed itself is also a bottleneck but it could be shown that floating point operations can be replaced by fixed point or integer arithmetics which allows the PDA processors to perform at considerable speeds and nearly equal translation quality. The problem that remains are the models' memory requirements. Clever encoding and compression approaches can decrease these requirements considerably. In particular language models can be encoded efficiently using the Bloomfilter data structure, a lossy hashing technique. The translation model (phrase table), however, cannot be easily compressed and phrase pairs have to be removed. The goal here is to remove enough phrase pairs to fit the model in the memory of a small, mobile device while maintaining the best possible translation performance. A new approach to removing phrase pairs is presented that estimates the relevance of a phrase pair based on actual usage statistics of this phrase pair when translating large quantities of data. The general idea of this approach was inspired by the Optimal Brain Damage algorithm for neural networks. Several options are investigated and we can show that the number of phrase pairs can be reduced by up to 80% without significantly affecting the translation performance. This outperforms all previously known phrase pair pruning techniques.

Improving Named Entity and Specialty Vocabulary Coverage The basic approach to gathering additional named entities and specialty vocabulary is to exploit other knowledge sources. In the first step we use a large medical database to improve the vocabulary coverage for medical translations which creates significant score improvements. In other domains, databases like this are not so readily available so name lists have to be manually generated. It is, however, not efficient to put a large vocabulary on a small device and it may interfere with the speech recognition performance of a speech to speech translation system. For this reason a personalization approach

is proposed that uses a background lexicon available through an online service. The mobile devices occasionally connect to the service and update their phrase tables based on the actual usage of the individual device and the encountered unknown words. Unfortunately, even with the added vocabulary it will never be possible to cover every word. A new approach is presented that uses monolingual encyclopedias and dictionaries to "communicate" unknown words. Instead of the actual unknown word, its definition is translated, which leads to considerable improvements in translation quality.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Automatic machine translation systems in research institutions have reached a considerable level of performance. Especially the statistical machine translation approaches have outperformed other earlier approaches since their introduction in the 1990s. Further research and the availability of more and more training data, faster processors and more computer memory allows researchers today to produce a translation quality that seemed impossible only a short time ago.

Particularly on limited domains like tourism, medical relief or force protection, the translations have already the potential to be very useful for real applications outside of laboratories or demonstrations. These domains are of specific interest, as the availability of human interpreters here is very limited or too costly, but a basic level of communication is necessary.

The question here is no longer how to get the technology to an acceptable level, but how the existing technology can be globally deployed. How can a deployable system be developed using the existing technology that a tourist, a doctor or a warfighter[1] can use and benefit from?

Table 1.1 shows some of the main differences between an existing laboratory system and a system that could actually be deployed.

The first difference between a research system and a deployable system is the number of generally supported languages. In research, most groups focus on a few language pairs for which large amounts of training data are available and do only occasional tests on other languages to confirm their

[1] "Warfighter" is a generic term used by the United States Department of Defense to refer to any member of the US armed forces. It is intended to be neutral regarding military service or branch, gender, and service status.

	Research system	Deployable system
Languages	Limited number	Many Language Pairs
Hardware	High-End Server	Notebook, Mobile Device
Application	Evaluations, Demonstrations	Actual Communication
Evaluation	BLEU/NIST Scores	Communication success
Users	Expert researchers	Tourists, Medical users, Military users
Interface	Complex, command line	Easy, interactive

Table 1.1: Qualitative differences between research systems and actually deployable systems

results. Systems for tourists, and especially for medical relief or military users, will have to support a much larger number of languages. It would be possible to market systems for tourists in a specific country and language that is popular, but it cannot be foreseen where medical relief situations or military tensions might develop.

In research labs, the translation systems are usually run on high-powered, expensive servers to be able to use larger and more complicated models and to realize even small performance improvements. The users are researchers and experts who do not necessarily care about the simplicity of the user interface. Their goal is to be flexible and to allow easy integration of additional components and a fast experimental turnaround. For non-research users, in particular in the mentioned tourism, medical and military applications, it is essential that the device is actually lightweight and mobile. A notebook computer might be a possibility in some situations, but generally the goal would be to have the translation system on a PDA, a handheld game console or even a cell phone. These users—tourists, medical staff and military personnel—will also not tolerate a complex user interface affecting their communication abilities and will demand an easy to use system.

Translation systems in research are usually applied in competitive evaluations or shown as demonstrations. The test sets in these situations are either designed to illustrate a specific capability or are standardized to allow

comparisons between different groups. The test sets do indeed try to emulate the actual task, but often fall considerably short of that goal, especially concerning the demand for named entities. This is supported by the evaluation metrics. Little weight is generally put on an individual named entity while this named entity could be essential for the actual communication success.

1.2 Scenarios

The domains of particular interest in this thesis are tourism dialogs, medical relief and force protection. Tourism dialogs are dialogs between a tourist and a local in a foreign country, usually initiated by the tourist. The tourist might want to book a flight, hotel room, rental car, ask for directions or restaurant recommendations; the local person will try to answer his questions (example in table 1.2).

	...
Tourist:	I would like to book a hotel room from Thursday until Saturday.
Hotel clerk:	Would you like a single or double room?
Hotel clerk:	A single room is 80 USD per night, a double room is 110 USD per night.
Tourist:	Double room, please. Can I pay with my credit card?
	...

Table 1.2: Tourism dialog

In a medical relief situation, medical personnel from another country visit an affected area to provide medical help. A translation system would then be used to communicate with affected locals—mainly doctor/patient dialogs during examinations. At first the patient will describe his problems and injuries, followed by the doctor giving instructions regarding medication and continuing treatments (example in Table 1.3). However, this application is not only limited to medical relief situations. A similar situation arises even within the United States, as hospitals are facing a large number of patients with limited English knowledge. This problem has become so significant that in August 2000, an Executive Order was released by US President Clinton requiring all hospitals to provide translation services for these patients, usually using interpreters (Executive Order 13166, August 2000).

Force protection generally refers to situations during military operations where the warfighters come into contact with the local population or local

	...
Doctor:	It's nothing too serious, we'll be able to help you with medication.
Patient:	I guess that's good to hear.
Doctor:	Let me explain. Your test results show you have a deficiency of vitamin B12.
Doctor:	The medical term is pernicious anemia, have you ever heard of it?
Patient:	I know what anemia is, but I don't know about this specific type.
Doctor:	Pernicious anemia is caused by the body's failure to absorb vitamin B12.
	...

Table 1.3: Medical dialog

law enforcement in a foreign country. This ranges from administrative tasks and dialogs related to civil engineering to the questioning of suspects. Conversations at checkpoints are also typical (example in table 1.4).

	...
Warfighter:	Can you open the hood of the car please?
Local:	Okay, I unlatched it. You can open it up now.
Warfighter:	Do you have any weapons or any dangerous materials in the car?
Local:	Dangerous weapons or hazardous materials, no.
Warfighter:	Okay, here is your ID back. Your car checked out, you can proceed.
	...

Table 1.4: Force protection dialog

In all situations, the translation system is generally assumed to be a two way speech to speech translation system, but this thesis will only discuss issues related to the translation of text. This assumes that the speech is already recognized and does not contain significant disfluencies. Issues related to speech synthesis will also not be discussed.

1.3 Thesis Outline

In the following chapter the current state of statistical machine translation will be introduced. Following that will be a detailed analysis in chapter 3 of how and why this falls considerably short and does not necessarily address the demands of the aforementioned users. Three main factors will be pointed out that are in the center of these concerns:

- Low cost portability for fast transfer to new language pairs
- Translation models for small, mobile devices
- Improving named entity and specialized vocabulary coverage

The following three chapters will then introduce various approaches and algorithms to overcome these problems separately for these three factors (chapters 4, 5, and 6). The last chapter 7 will give additional conclusions and potential directions for future work.

Chapter 2

Statistical Machine Translation

2.1 Introduction

This section will generally introduce statistical machine translation theory. Alternative approaches have been proposed, but statistical machine translation has consistently outperformed all other methods in competitive evaluations (Akiba et al., 2004; Eck and Hori, 2005; Paul, 2006; Fordyce, 2007). High quality tools to train and use statistical machine translation systems are readily available (Koehn, 2004; Koehn et al., 2007). This leads to short development times of baseline translation systems and makes this technique the method of choice for many research groups worldwide.

Alternative machine translation approaches include Example-based machine translation (EBMT) (Sato and Nagao, 1990; Brown, 1997) and transfer-based machine translation (Lavie et al., 2004). Transfer-based machine translation analyzes the source text into a syntax tree and then converts the tree into the form required by the target syntax (for example, moving the verb complex). An alternative approach is to analyze the source text into some formalism that is intended to capture meaning, not just the grammatical form, called an *interlingua*. It was also proposed to use common languages as an interlingua or *pivot* language (Reichert and Waibel, 2004; Babych et al., 2007).

Recently, statistical machine translation has been augmented with syntactic (Yamada and Knight, 2001; Chiang, 2005; Venugopal et al., 2007) and semantic (Cherry, 2008) information. While this can improve the results, it is not easily applicable to all language pairs as syntactic and semantic parsers are necessary and not available for many languages. This will also require a higher computational effort.

2.2 Statistical Machine Translation

2.2.1 General Approach

Given a source sentence $s_1^J = s_1 s_2 \dots s_J$, which is to be translated to a target sentence $t_1^I = t_1 t_2 \dots t_I$. The goal is to choose the target sentence with the highest probability given this source sentence.

$$\hat{t}_1^I = \arg\max_{t_1^I} P(t_1^I | s_1^J))$$

The original approach introduced in Brown et al. (1990) and Brown et al. (1993) uses a noisy channel model. Using Bayes rule, the probability $P(t_1^I|s_1^J)$ can be written as $\frac{P(s_1^J|t_1^I)*P(t_1^I)}{P(s_1^J)}$. As the probability of the source sentence remains constant in the comparison among the same string, the overall term becomes:

$$\hat{t}_1^I = \arg\max_{t_1^I} P(t_1^I|s_1^J) = \arg\max_{t_1^I} P(s_1^J|t_1^I) * P(t_1^I)$$

This allows the clear separation of two models: the language model $P(t_1^I)$ and the translation model $P(s_1^J|t_1^I)$.

An alternative was introduced in Och and Ney (2002). Here, the posterior probability $P(t_1^I|s_1^J)$ is directly modeled in a log-linear framework using a set of M feature functions $h_m(t_1^I|s_1^J), m = 1...M$. Each feature function has a scaling factor λ_m. The probability is then given by:

$$P(t_1^I|s_1^J) = \frac{exp[\sum_{m=1}^M \lambda_m h_m(t_1^I|s_1^J)]}{\sum_{t'^I_1} exp[\sum_{m=1}^M \lambda_m h_m(t'^I_1|s_1^J)]}$$

The goal is to find the translation that maximizes the combined models:

$$\hat{t}_1^I = \arg\max_{t_1^I} \sum_{m=1}^M \lambda_m h_m(t_1^I|s_1^J)$$

The main advantage is that the log-linear framework allows for the easier integration of additional models beyond the language and translation models, like distortion models and syntax or dependency based models.

The parameters for these models are generally estimated from monolingual and bilingual training data. Standard language models are trained from monolingual corpora in the target language, while the translation models are trained from parallel bilingual corpora in the target and source language. A parallel corpus is sentence aligned, so the translations are known on a

sentence level. Generally, larger corpora lead to better model estimates and, in turn, better translation results (Ueffing and Ney, 2002). However, the domain of the corpora should be close to the domain of the test sentences. Adding data from significantly different domains often does not improve the translations.

2.2.2 Translation Models

The goal of the translation model is to estimate the probability of a word to word or phrase[1] to phrase translation. Most methods used to extract phrase translation tables from a parallel corpus start with a word alignment.

A manual alignment of an example Spanish/English sentence pair is shown in figure 2.1

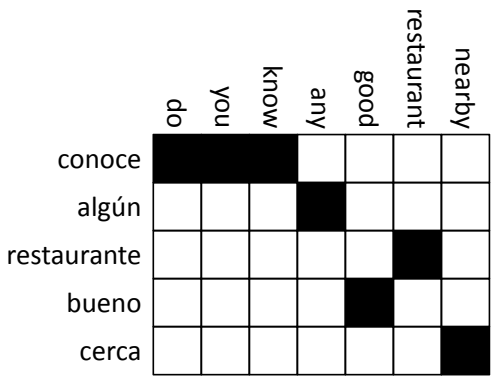


Figure 2.1: Manual word alignment

Standard word alignment models are the IBM Models 1 through 5 (Brown et al., 1993) (often abbreviated to IBM1, ..., IBM5) and the HMM alignment model (Vogel et al., 1996). These models try to reproduce the manual alignments using more and more complex models. IBM Model 1 only considers the co-occurrence of words and does not use any position or fertility information which is introduced in the higher IBM Models. An automatic alignment result for the example sentence pair could be similar to figure 2.2. Here, the size of the circles indicates the probability of the respective word pair alignment. This correspondence matrix already allows the extraction of word to word translation pairs.

However, a simple word to word translation cannot capture local reorderings. For example, English places the adjective in “good restaurant” before

[1]The term *phrase* in statistical machine translation only refers to a group of words in sequence and does not have a syntactic or linguistic meaning.

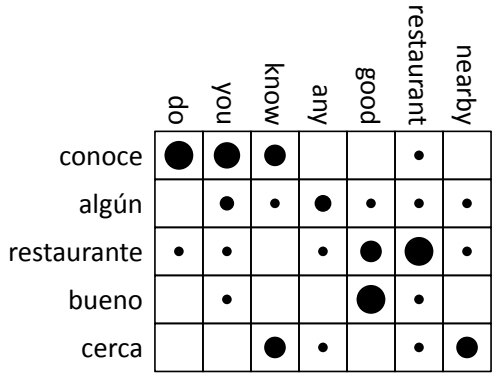


Figure 2.2: Automatic word alignment

the noun, while Spanish places it after the noun in "restaurante bueno". Word to word translation also does not allow for single words to be translated by multiple words like "conoce", which is equivalent to "do you know". For these reasons, phrase to phrase translation was introduced. Here, the goal is to extract phrase pairs from the training data that can capture longer phrases and handle local reordering and other issues.

Most phrase extraction methods are based on the word alignment information. Various algorithms for phrase pair extraction have been proposed, such as Koehn et al. (2003), Vogel (2005) and Zhao and Waibel (2005).

The PESA phrase alignment, introduced in Vogel (2005), will be used in some of the later experiments. This phrase alignment is traditionally based on IBM Model 1 word alignment probabilities, but other word alignment models can be used as well. Given a source phrase, the PESA alignment tries to find the target phrase that will maximize the combined probability of the source and target phrase splits being translation pairs. This is indicated in figure 2.3. The goal is to maximize the probability in the "inner" (dark gray) and "outer" (white) parts of the phrase pair areas and minimize the probability in the unaligned (gray) areas.

In this case, the optimal split leads to the phrase pair "restaurante bueno" → "good restaurant". The same is done across all sentences and for each phrase pair the top translation candidates are collected. Various feature scores are assigned to each phrase pair, but the scores can be combined to form a translation probability. The extracted phrase pairs could be similar to the examples in table 2.1.

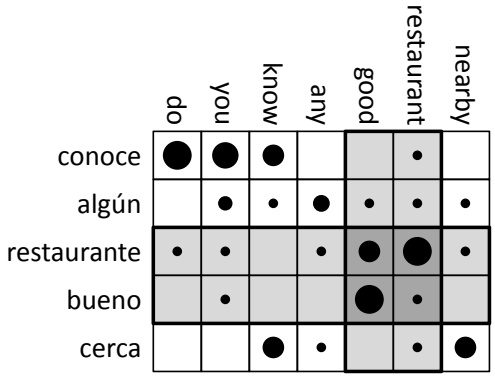


Figure 2.3: PESA phrase alignment

Source Phrase	Target Phrase	Probability
	...	
restaurante	restaurant	0.4
restaurante	the restaurant	0.2
restaurante	steakhouse	0.2
restaurante	in the restaurant	0.1
	...	
restaurante bueno	good restaurant	0.5
restaurante bueno	the good restaurant	0.2
restaurante bueno	good restaurant in	0.1
	...	

Table 2.1: Phrase pair examples

Online vs. Offline Phrase Tables Two main methods exist to train and apply phrase tables. In the *offline* case, the phrase pairs are pre-extracted from the training data and stored in a phrase table file that is later used in decoding. In this case, phrase pairs are extracted for phrases up to a certain length. This is based on the phrase distribution in the bilingual training corpus with frequency thresholds for higher phrase lengths. It is common to extract phrases for all unigrams, bigrams and trigrams and use frequency thresholds of up to 4 for phrases up to length 10. This means phrase pairs for a phrase of length 10 are only extracted if the phrase occurred at least 4 times. For longer phrases, the probability of the phrase to occur in the test data will generally be too low.

An alternative approach to phrase extraction is *online* extraction from the bilingual training data instead of having a pre-extracted phrase table.

Online extraction dynamically extracts phrase pairs necessary for the actual test sentence as proposed in Callison-Burch et al. (2005) and Zhang and Vogel (2005). This technique usually improves the performance, as arbitrarily long phrases can be matched. However, the necessary computing power is much higher compared to pre-extracting the phrase pairs. This currently prohibits the application of this technique on small devices.

2.2.3 Language Models

The language model is the second model in the classical approach and one of the models in the log-linear approach. This model tries to estimate the probability $P(t_1^I)$ of a phrase or sentence in the target language regardless of the source sentence. This probability is generally decomposed using conditional probabilities depending on the word histories.

$$P(t_1^I) = P(t_1) * P(t_2|t_1) * \ldots * P(t_k|t_1 \ldots t_{k-1}) * \ldots * P(t_I|t_1 \ldots t_{I-1})$$

The history is typically limited to 2 to 5 words. If the history is limited to (n-1), $P(t_k|t_1 \ldots t_{k-1})$ is estimated by $P(t_k|t_{k-(n-1)} \ldots t_{k-1})$, which is called an n-gram language model.

N-gram language models are trained from monolingual corpora in the target language. Standard implementations are the SRI language modeling toolkit (Stolcke, 2002) and suffix array based implementations (Zhang and Vogel, 2006). Suffix array implementations allow arbitrarily long histories. The whole training corpus is kept in memory and the n-gram frequencies are extracted as needed from a suffix array data structure.

However, not all histories might have been seen, and the language model has to discount probability from seen events and assign to unseen events with discounting and smoothing methods. Standard language model discounting and smoothing methods are Good Turing and Kneser Ney (Kneser and Ney, 1995; Chen and Goodman, 1996).

Other proposals for language models and improved techniques have been made, but the success in statistical machine translation has been limited up to now (Rosenfeld, 2000; Raab, 2006). These techniques did have more success in other natural language processing tasks like automatic speech recognition.

2.2.4 Decoding

After the model training is finished, the models are applied to translate source sentences in a process called *decoding*. The decoding is divided into two steps:

- Building the translation lattice
- Best path search

The translation lattice is a graph structure and is built by starting from the source sentence in a linear graph, with each source word representing one edge. The trained phrase table is then applied to this graph. The decoder searches for matching source phrases and adds additional edges representing the respective target phrase. Figure 2.4 shows a possible lattice for the earlier example sentence. Each path through the lattice represents one translation hypothesis. The hypotheses also store coverage and backtracking information as well as the translation model and other scores.

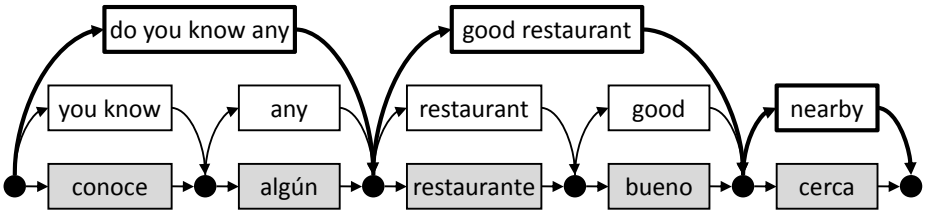


Figure 2.4: Lattice example

The number of hypotheses is usually constrained by a beam factor or other measure to limit the computational complexity and search space.

In the second step, the decoder searches for the best of these hypotheses by calculating the score for additional models like the language model and sentence length model. The decoder is also able to eliminate a hypothesis early if the estimated score will be below a certain threshold (see also Ueffing et al. (2002) and Och et al. (2001)).

If no reordering model is applied, the decoding is called *monotone*. In monotone decoding, reordering is limited to the local reordering realized by phrases. A reordering model is applied in *non-monotone* decoding. Considering all word permutations is computationally expensive, so it is generally limited to a reordering window (Vogel, 2003). Other reordering strategies can jump ahead in the source sentence and leave certain words for later translation. Additional methods reorder words and phrases based on syntactical parse information (Elming, 2008).

Generally, word reordering can be computationally expensive, but it can also offer significant performance improvements compared to monotone decoding. This is particularly the case if the source and target languages have very different word order.

Minimum Error Rate Training The growing number of models being used in decoding makes it difficult to manually determine the optimal scaling factors for each model. While this was possible when only two models were used, modern statistical machine translation uses a minimum error rate training process. Minimum error rate training automatically finds optimized scaling factors based on a set of test sentences with known reference translations (Och, 2003; Venugopal et al., 2005).

2.3 Evaluation of Translation Quality

It is very important to be able to measure the quality of a produced translation in order to determine which approach gives the best improvements over a baseline system. Evaluation metrics also allow a general estimation of the overall performance and comparisons to a human translator. There are generally two ways to evaluate machine translation quality.

2.3.1 Subjective Evaluation

In subjective evaluations, human evaluators are asked to rate the translations. A common way is to ask evaluators to rate sentences according to fluency and adequacy on a five point scale. Fluency measures the grammatical correctness of a given translation in the target language, regardless of the source sentence (Usually the source sentence is not seen by the evaluator). Adequacy measures how well the information in the source sentence was transferred to the target sentence. This technique was used in the IWSLT evaluation campaigns. For IWSLT 2005 an additional measurement "meaning maintenance" was introduced. While similar to adequacy, evaluators were explicitly instructed to give lower scores to potentially misleading information, even if other parts of the sentence were well translated (Akiba et al., 2004; Eck and Hori, 2005; Paul, 2006; Fordyce, 2007).

In another approach the human evaluators are asked to manually correct the machine translation output. After the edits have been done, it is automatically measured how many edit operations were necessary. This uses the Translation Error Rate (TER) (Snover et al., 2006) metric. With the human correction input, the evaluation metric is then called HTER.

Question based subjective evaluation was proposed in Voss and Tate (2006) and Jones et al. (2007). Here the evaluators are asked to answer question based on the translations. The overall goal is to get closer to an evaluation method that is used for language students in order to be more comparable to human experts. These methods also simplify the evaluators task. It is hard to define the difference between an adequacy of 2 or 3, and this might heavily depend on the individual evaluator. On the other hand, it is a more clearly defined task to answer a question based on a translation (as in Voss and Tate (2006) and Jones et al. (2007)) or correct a machine translation output as in HTER.

Unfortunately, these subjective evaluation procedures are very expensive and time consuming.

2.3.2 Automatic Evaluation

For this reason, automatic evaluation procedures have been proposed. It could be shown that they can offer a high correlation with subjective evaluations, while being much cheaper and producing evaluation results in just a matter of seconds.

The most popular automatic evaluation metrics for machine translation are BLEU and NIST scores.

BLEU (Papineni et al., 2001) measures the precision of 1-grams to 4-grams of the translation output compared with the closest reference translation and calculates their geometric mean. BLEU uses a brevity penalty to penalize too short translations as a substitute for recall. BLEU usually correlates very well with human fluency judgments, while the correlation with adequacy is more limited. BLEU scores range between 0(worst) and 1(best)[2].

The NIST score (Doddington, 2001) introduces a notion of information gain. This means that a translation gets a higher benefit for correctly translating rare words and n-grams than it gets for common ones. It can be assumed that the rare words and n-grams are content words and, thus, more important for the overall translation quality than a common function word. This leads to the result that the NIST score is usually better correlated to adequacy than to fluency. NIST scores are not normalized, so they are positive numbers with 0 being the worst possible score.

A problem with the BLEU and NIST scores is that they only consider exact matches. This means a word has to exactly match one of the references, otherwise it will not be considered correct. Even words with only a slight morphological difference will be considered incorrect. The METEOR score

[2]Occasionally BLEU scores are reported as percentages, then ranging from 0 to 100

(Banerjee and Lavie, 2005) tries to circumvent this problem. It uses different stages of matching words. Starting with the exact match as in BLEU and NIST, it also considers stem matches and synonym matches.

The common metrics in speech recognition are word error rate (WER) and position independent error rate (PER), and these are also occasionally used to measure machine translation performance (Niessen et al., 2000).

Despite its shortcomings, BLEU is still the most commonly used metric in statistical machine translation, so all automatic evaluations in this thesis were done using BLEU. Some of the related publications also report NIST scores. BLEU scores will generally be reported with four digits. However, it is important to be aware that differences in the third and fourth digits are often not statistically significant. To measure the statistical confidence of the scores and score improvements, the method proposed in Zhang and Vogel (2004) is used.

It is common procedure in most speech to speech translation systems to convert all data to lower case and remove all punctuation marks, which was also done in all reported experiments. For easier readability the example translations in this thesis are shown in mixed case with inserted punctuation marks.

Chapter 3

Research System vs. Deployable System

3.1 Introduction

As it was pointed out in chapter 1, machine translation systems on limited domains—specifically tourism, medical relief, and force protection—have reached a considerable level of performance.

Some examples from the output of the UKA/CMU[1] translation system for IWSLT2006[2] (Eck et al., 2006) are shown in table 3.1 (tourism domain translations from Japanese).

Reference:	I would like to rent a car.
Translation:	I want to rent a car.
Reference:	I did not make a reservation. Can I still rent one?
Translation:	I don't have a reservation. Can I rent?
Reference:	I would like to rent this car for a week or so.
Translation:	About this I want to rent a car for a week.
Reference:	This movie will last for about two hours from eight o'clock to ten o'clock.
Translation:	This movie is two hour from eight to ten.

Table 3.1: Example translation from IWSLT 2006 (Japanese → English)

The translations are not perfect and certainly contain various mistakes, but

[1] UKA: Universität Karlsruhe; CMU: Carnegie Mellon University

[2] IWSLT: International Workshop for Spoken Language Translations

it would still be very useful for a tourist in a foreign country to be able to accomplish this level of communication.

For medical and military applications, the requirements might be higher as translation errors could be far more serious. This was discussed in Flores et al. (2003) and Neergard (2003). Misunderstanding the recommended dose of a medicine or misunderstandings that occur during an interrogation could lead to difficult situations. However, even in these domains, not all tasks are this critical and a functioning translation system could definitely be useful.

The goal of the TransTac project, funded by the Defense Advanced Research Projects Agency (DARPA), is to rapidly develop speech to speech translation systems for the US military. In military situations, an automatic translation system eliminates certain risks that could occur when using a local interpreter who might actually be opposed to the military goals or might be in danger by working with the warfighters. The survey and informal comments made by the warfighters during the TransTac evaluations indicate the same points. The majority would use the presented speech to speech translation systems, but they acknowledge potential problems in critical situations (Schlenoff et al., 2007). As they only infrequently have access to a human interpreter, they see the translation systems as a useful tool to allow them to communicate with locals at least on a basic level. They also recognize the risks with human interpreters and would prefer an automatic system they can trust to be completely neutral.

Nevertheless, translation systems are not deployed on a large scale and still have certain problems. The question that remains is:

Where does the technology still fall short? What is lacking that would enable systems to actually be deployed on a larger scale?

The following sections will analyze these questions in detail. The main differences listed in table 3.2 were already informally mentioned in the introduction chapter 1. These differences will be discussed in the following four sections.

The limited number of supported languages and the hardware differences will be discussed in sections 3.2 and 3.3. The applications and evaluation differences are only a symptom of the underlying problem and it will be argued in section 3.4 that the solution is the extended support of named entities and specialty vocabulary. The disparity between the expert and non-expert users and the interfaces they prefer are closely related and will be discussed in section 3.5.

	Research system	Deployable system
Languages	Limited number	Many Language Pairs
Hardware	High-End Server	Notebook, Mobile Device
Application	Evaluations, Demonstrations	Actual Communication
Evaluation	BLEU/NIST Scores	Communication success
Users	Expert researchers	Tourists, Medical users, Military users
Interface	Complex, command line	Easy, interactive

Table 3.2: Qualitative differences between research systems and actually deployable systems

3.2 Supported Language Pairs

3.2.1 Limited Number vs. Many Language Pairs

The first issue that prevents the available technology from being applied on a larger scale for actual users is the number of supported languages. Research systems usually focus on a small number of language pairs. The NIST evaluations (`www.nist.gov/speech/tests/mt`) have traditionally focused on Chinese → English and Arabic → English. The most popular evaluation campaign for spoken language translation, the International Workshop on Spoken Language Translation (IWSLT), has supported 5 language pairs since its inception in 2004 (see table 3.3 and IWSLT overview papers Akiba et al. (2004), Eck and Hori (2005), Paul (2006), and Fordyce (2007)).

Up to now the TransTac project focused on the language pair Iraqi-Arabic ↔ English with an additional experiment on Farsi ↔ English in mid 2007. A small number of other languages is also researched in the InterACT[3] lab, specifically Spanish and German.

[3]International Center for Advanced Communication Technologies, joint research lab at the Universität Karlsruhe (TH) in Germany, Carnegie Mellon University in Pittsburgh, USA and The Hong Kong University of Science and Technology, (`www.is.cs.cmu.edu`)

Evaluation	Languages (translated into English)
IWSLT 2004	Chinese, Japanese
IWSLT 2005	Chinese[4], Japanese, Korean, Arabic
IWSLT 2006	Chinese, Japanese, Italian, Arabic
IWSLT 2007	Chinese, Japanese, Italian, Arabic

Table 3.3: Languages in the IWSLT evaluation campaigns

Nevertheless, this is still not comparable to the estimated 7,000 languages that exist worldwide, with over 250 spoken by more than 1 million people (Gordon, 2005).

Reasons The main reason for this situation is the limited availability of training data for statistical machine translation systems. Statistical machine translation relies on parallel bilingual training corpora to extract translation pairs, and such corpora are only available for a small number of language pairs. For example, the IWSLT evaluation campaign uses the BTEC corpus (Takezawa et al., 2002) and this corpus is only available in the languages listed in table 3.3, with the addition of very few others. BTEC is one of the few corpora that was specifically designed as a corpus for statistical machine translation, whereas other bilingual corpora have resulted as a byproduct of other efforts. Some examples of this include newspapers and newswires in China that publish articles in English and Chinese, and organizations like the United Nations and the European Union that translate their protocols to the major languages of their member states. The EuroParl (Koehn, 2005) and Hansard (Canadian Parliament, French-English) corpora are examples here.

In addition to this, researchers generally implicitly or explicitly assume that research advances made on a specific language will carry over to another language pair. It is at least viable to assume that related languages will pose similar challenges, like differences in word order or morphology, that could be overcome with comparable approaches. Research also focuses on certain languages, as some experience in these languages is already available and a new language might pose a more difficult venture.

For actual users, the limitation of languages is certainly not acceptable. For example, medical relief efforts after the tsunami of December 2004 took place in Indonesia, Thailand, Sri Lanka, and India (among others). These countries have a wide variety of languages, which posed great difficulties

[4]IWSLT 2005 also offered a track translating English to Chinese

for humanitarian aid workers. According to Gordon (2005) there are 737 languages spoken in Indonesia alone. Hardly any of these languages are in the research focus right now. Immediate coverage of these languages and very short term deployment of translation systems might be impossible, but even after a couple of months of preparation, the availability of translation devices could still be beneficial.

Military users also face a similar problem with significant US military personnel stationed in Afghanistan, Iraq, and various other countries. Afghanistan specifically is home to a variety of languages and regional dialects that create a great hardship for communication.

This does not mean that military and medical users require support for a large number of language pairs immediately, but there will be sudden demands for a specific languages and translation systems have to be rapidly developed and deployed.

3.2.2 Portability to New Languages

The central issue in building a statistical machine translation system in a new language is the availability of a significant amount of bilingual training data (see chapter 2). The solution that first springs to mind is to have human translators translate large corpora to the new language, but this process is very cost intensive and time consuming. However, current technology requires parallel bilingual training data, so the necessary human translations should be most effectively used. This means giving precedence to translating those sentences into the new language that will be most advantageous for the performance of a statistical machine translation system. Chapter 4 will introduce a number of approaches to this problem that will show how time and cost can be effectively decreased while still generating a valuable bilingual training corpus. This is accomplished by pre-estimating the value of a translated sentence according to various statistics. The sentences can then be sorted according to their estimated value and the top sentences can be chosen for translation.

The availability of bilingual corpora is, however, not the only issue in building a translation system for a language pair. Each language poses specific difficulties, for example in encoding, script, word segmentation, and vowelization. Many languages like Iraqi-Arabic do not even have a well defined written form. All these problems are extremely dependent on the individual language, so individual solutions will be necessary. For this reason these issues were not considered in this thesis.

3.3 Computing Hardware

3.3.1 High End Servers vs. Mobile Devices

Translation systems in research labs are usually run on high-end server machines with large amounts of memory and fast, often multi-core, processors. One reason for this is that these machines allow a faster experimental turnaround. The machines support easier development of new models without having to be too concerned about memory or processor usage. The sizable computers also allow the fast tuning of translation systems to gain even the tiniest bit of performance improvement. The general goal here is to quickly translate as many sentences as possible, to try different approaches and get convincing results for the applied methods.

For actual users, however, it is far more important to have a mobile device that they can easily carry. A PDA, a handheld game console or even a cell phone would be more widely accepted by tourists. For military or medical relief users, a notebook computer might be tolerable in some situations. Generally, users will certainly be willing to trade slight performance drops for easy mobility and reasonably fast translations. For example, a specific reordering model might only gain a fraction of one point in BLEU score, and it might actually multiply the translation time and memory consumption. However, one point in BLEU score could make a very big difference in competitive evaluations like NIST or IWSLT, so the added translation time is willingly accepted for a research system. An actual user, on the other hand, might not even notice the quality difference and would undoubtedly prefer the faster translation that could potentially run on a much smaller device without using the reordering model.

The TransTac project specifically takes a first step in this direction as the developed speech to speech translation systems have to be lightweight and mobile. Translation delays will also heavily affect the overall performance, as the groups are evaluated based on the number of "concepts" that are communicated within a certain time frame (among other metrics). All groups opted for high-end notebooks as they provide nearly the same computing power as desktop and server machines, but experiments have also included PDAs. Notebooks might be an acceptable compromise for certain situations in military or medical relief efforts, but no tourist can be expected to carry a notebook with him. The added cost to acquire a notebook for this would also probably outweigh the perceived benefits. A PDA may offer a compromise, as they are smaller and often more affordable compared to a notebook computer. Nevertheless, the best possible situation would be to have the translation system running on a cell phone. A large percentage of people in industrialized

countries already owns a cell phone and would not need to carry an additional device. Many newly developed cell phones offer greatly enhanced capabilities, equal or close to PDAs (smartphones). Table 3.4 lists a number of selected lightweight and mobile devices.

HTC Touch Pro	
Memory	288 MB
Processor	Qualcomm MSM7201A @ 528 MHz
RIM Blackberry Curve 8330	
Memory	96 MB
Processor	Qualcomm MSM6550 @ 225 MHz
Apple iPhone	
Memory	128 MB
Processor	ARM 1176 @ 412 Mhz
Sony PSP	
Memory	32 MB + 4 MB
Processor	MIPS R4000-based @ 333 MHz
Nintendo DS	
Memory	4 MB
Processor	ARM946E-S @ 67 MHz + ARM7TDMI @ 33 MHz

Table 3.4: Examples for lightweight, mobile devices

The Touch Pro is currently (August 2008) the top model from HTC while the Curve 8330 is an example for a recent Blackberry model from RIM. The popular iPhone from Apple and the hand held game consoles Sony PSP and Nintendo DS are also listed (All data from the manufacturers websites `www.htc.com`, `www.apple.com`, `www.rim.com`, `www.sony.com`, `www.nintendo.com`). Data storage is generally not an issue, as all devices offer extensible storage via flash memory cards.

Comparing these numbers with a standard server/desktop computer and even a notebook shows the differences in available memory and computing power. Current notebooks often offer more than 2 GB of memory and use dual or quad-core processors at speeds of well over 2 GHz (current notebook price about 1200-1500 USD). A top PDA/smartphone will cost close to 1000 USD (without contract), while the hand held game consoles are currently priced at 130 USD (Nintendo DS) and 170 USD (Sony PSP).

3.3.2 Building Mobile Systems

The main issue in making the translation systems run on a PDA are the memory requirements of the translation and language models. The processor speed itself is also a bottleneck, but it could be shown that floating point operations can be replaced by fixed point or integer arithmetics which allow the PDA processors to perform at respectable speeds and nearly equal translation quality (Zhang and Vogel, 2007; Hsiao et al., 2006).

What remains a problem are the models' memory requirements. Clever encoding and compression approaches can decrease these requirements significantly. In particular language models can be encoded very efficiently using the Bloomfilter data structure, a lossy hashing technique, and can usually be cut to manageable sizes (Talbot and Osborne, 2007a,b). The phrase table cannot be so easily compressed, and phrase pairs will have to be removed to meet the size requirements. The goal here is to remove enough phrase pairs to fit the model in the memory of a mobile and lightweight device while maintaining the best possible translation performance. Approaches to identify phrase pairs that can be removed and have the least impact on the overall translation performance are presented in chapter 5.

3.4 Named Entities and Specialty Terms

3.4.1 Limited Named Entity Support

The third major issue that prevents research translation systems from finding their way to actual usage areas is the missing support of a large number of named entities and other specialty vocabulary. Table 3.5 shows some examples of sentences with named entities from the IWSLT 2006 evaluation and the translations of the CMU/UKA system (Eck et al., 2006).

Reference:	My name is Xiao Bai. This is my driver's license. Please take a look.
Translation:	Please look. Here's my driving license.
Reference:	Nanjing Road in The Bund is the busiest spot here.
Translation:	This is the most lively place.
Reference:	You can embark at the Wu Song Harbour.
Translation:	Could you steward Pier.

Table 3.5: IWSLT 2006 example sentences with named entities

The translation quality is, again, mostly acceptable with the exception of the names that are missing. Unfortunately, the missing names make the translations unusable for any real user.

The reason why these names are missing is simple. The training data that was allowed for the IWSLT 2006 evaluation did not contain these names. This also means that every group could assume that no other group would translate those names correctly so the missing names would not affect the group's ranking.

As BLEU scores are the main automatic evaluation metric, the names also become less relevant as each name in table 3.5 occurs only once. This means fixing the translation of one of the names would only result in an insignificant BLEU score change, much less than other improvements that could affect more sentences. The BLEU score might even value names lower than other words. For example given the reference sentence "When is my flight to Tokyo" missing the name "Tokyo" will result in a higher BLEU score for this sentence than missing the "to". This is the case as all words are valued equally in the BLEU score and "Tokyo" occurs in only one bigram, trigram and 4-gram while "to" occurs in two bigrams, trigrams and 4-grams.

It should be noted that BLEU scores were not designed to score individual sentences and generally do give better correlation to human judgments on larger test sets. Alternative scoring methods like NIST incorporate information gain and would value a content word higher than a function word. Most competitive evaluations also use human judgments of the translations on at least some of their translation tracks, and a human will certainly value the named entities and content words.

Generally all named entities will be low frequency and only make up a small percentage of the overall tokens, but the names have a high impact on communication success as they carry a significant part of the information. A sentence might often only contain one named entity as in most of the examples in table 3.5, but missing this named entity makes it very difficult or even impossible to achieve communication success. Martine Adda-Decker and Lori Lamel list vocabulary sizes for various word categories in chapter 5 of Schultz and Kirchhoff (2006). Function words are in the hundreds, general and technical content words are in thousands of word types. Comparing this to the estimated millions of named entities illustrates the overall problem. Correct translations for function and common content words are certainly necessary as well and should be further improved in the future. However, the focus in this thesis will be the improvement of the named entity and specialty vocabulary coverage.

The named entity coverage of the BTEC corpus is particularly limited. The BTEC corpus consists of Japanese phrase books for tourists, which

means that it mainly contains named entities from Japan and very few names from other countries. However, phrase books tend to give only example names which leads to the fact that the standard examples are frequently present, while rare names are never used. This behavior was specifically investigated for the city names in Japan, checking the number of occurrences of 144 relevant city names listed at `www.japan-guide.com` in the complete English/Japanese BTEC corpus (about 1 million words on the English side). Only 49 of these names, listed in table 3.6 occur in the BTEC corpus while the other 95 never occur. Table 3.6 shows the disparity in the number of occurrences. Tokyo, with 672 occurrences, has nearly twice as many as all other cities combined and only 11 city names occur 10 or more times.

City	Frequency in BTEC	City	Frequency in BTEC
Tokyo	672	Hiroshima	2
Narita	64	Kanazawa	2
Kyoto	48	Matsuyama	2
Osaka	45	Nagasaki	2
Ube	29	Sendai	2
Fuji	21	Takamatsu	2
Yokohama	20	Toyota	2
Yamaguchi	17	Ashiya	1
Nara	11	Fukuoka	1
Chiba	10	Funabashi	1
Mito	10	Hofu	1
Kamakura	9	Ichihara	1
Kobe	8	Inuyama	1
Asahi	7	Kashiwa	1
Nagoya	7	Kawasaki	1
Sapporo	6	Matsumoto	1
Nagano	5	Morioka	1
Nikko	5	Niigata	1
Hakone	4	Okayama	1
Otsu	4	Saitama	1
Oyama	4	Takasaki	1
Sakai	3	Takayama	1
Seto	3	Toyama	1
Akita	2	Zama	1
Hamamatsu	2		

Table 3.6: Japanese city names in the BTEC corpus

It is also important to keep in mind that a Japanese ↔ English translation system could be used by Japanese speaking tourists in an English speaking country and city names from these countries are very rare in the BTEC corpus. Parts of the BTEC corpus also exist in other languages, such as Korean or Arabic. Here, the local names are completely missing, as these are merely translations from the English BTEC sentences.

This problem does not only affect city names, but other named entities and specific terms that will be important, like food items, general locations, street names, etc. Military users may have similar needs, and medical applications are especially known for having a wide variety of technical terms and medication names.

It is important to note that every user within one of the three groups will have their own specific needs. A tourist might have some specific hobbies or interests that he or she would like to research in a foreign country. They may also have food or shopping preferences. Nearly every medical or military professional has some kind of specialization, that heavily influences his vocabulary and phrase usage. Continuing with the city name example from above, it can be assumed that not every tourist will need all city names, but every tourist will visit a different area and will need some subset of the cities and locations. A standard machine translation system trained on the BTEC corpus would not completely fulfill this need right now.

3.4.2 Supporting Named Entities

The general task is to improve the support and coverage for named entities. Just adding more standard bilingual data will not solve this problem. It would take a very large amount of data to cover a high percentage of named entities and this would not be very efficient. The comparably small number of function and common content words are likely to be already well covered as illustrated in table 3.5 and table 3.1. It is also not clear which sources could be useful here because phrase books will always focus on the most common names. The only option would be travel guides that cover a certain country or area, but also these will not accomplish this efficiently.

Additional problems arise with the millions of person names and the fact that new named entities are continuously added.

The first goal is the possibility to separately collect named entities and integrate them in existing systems. The lack of memory in the mobile devices might also make it necessary to personalize a subset to the specific need of the individual user. This subset also supports the ASR performance, as a large vocabulary is especially hard to handle in speech recognition.

In addition to this, it will be necessary to have some kind of backup

strategy in place as it will not be possible to collect *all* named entities and specialty vocabulary. Right now if a word remains unknown it will be skipped and not translated as seen in table 3.5. This can be acceptable if the character sets of the languages are identical or close, but will not be satisfactory if the character sets are different. The translation should at least offer some way to handle these words. Chapter 6 will present approaches to solving these problems.

3.5 Interface and Users

The final differences between a research and an actually deployable system are the users and the user interface. In a lab setting, the main users are experts and researchers, and their goal is to effectively test new approaches and be very flexible when using a translation system. They typically use the same test set over and over again to be able to measure the differences between different settings and evaluate the improvements. The user interface is usually a command line with various parameters, parameter files and support programs.

This would certainly be completely unacceptable to actual users. Users demand an easy interface that will interfere as little as possible with the ongoing communication, and they will not be open to a system with a complicated interface. Interfaces like this have been researched and developed in different variations. Generally, users will prefer to actually speak to the system and get audible output. This primarily requires adding ASR and speech synthesis components for both languages. A display will mainly serve to check the correctness of the input and for status messages.

A simple graphical interface on a PDA is shown in figure 3.1. The display shows the output for both languages. On the English side, the first line is the ASR output which is translated to Japanese and shown in the Japanese section. The second line in the English part is the "back-translation", a second translation of the Japanese translation back to English. This can be useful to double-check the translation for the English speaker as it can be assumed that the translation will be correct if the back-translation is correct. However, the back-translation can underestimate the performance and can be confusing for non-expert users. Two buttons on the PDA are used as push-to-talk buttons for English and Japanese respectively.

The TransTac project developing speech to speech translation systems for military users required a hands-free and eyes-free system. That means that no display is permitted and no button can be pressed to communicate using the system. This was developed at InterACT as described in

Bach et al. (2007), but the actual users did prefer a push-to-talk button in order to have control over what gets translated, so the hands-free requirement was effectively dropped later.

As usable interfaces are available these issues will not be discussed in later parts of this thesis.

Figure 3.1: PDA interface

3.6 Overview

As pointed out in the previous sections, the main part of this thesis consists of three chapters corresponding to the three main issues identified and discussed here: Chapter 4 will present approaches to effectively support additional languages at low cost, chapter 5 will discuss solutions for using translation systems on small devices with limited resources and chapter 6 will develop approaches to improve the handling of named entities and specialized vocabulary in addition to personalization issues.

3.7 Related Work

Each of the three main chapters will separately cite relevant related work. However, some related work that is not specifically relevant to a certain chapter but related to the overall problem of developing speech to speech translation systems will be discussed here.

Generally various speech to speech translation systems have been developed over time e.g. Lavie et al. (1997). This system was still running on a larger computer and not mobile. Language portability became an issue later e.g. in Black et al. (2002). Specifically designed for the tourism domain is the translation system by NEC Research presented in Isotani et al. (2002) and Isotani et al. (2003).

An alternative approach is presented in Yamabana et al. (2003). Here the mobile devices communicate via wireless connections with a server computer. This allows for a better translation performance, but requires a constant connection.

The question of how to rapidly port the systems to new languages were also explored in Black et al. (2002) for the language pair Croatian $\leftrightarrow$ English and in Engelbrecht and Schultz (2005) for the language pair Afrikaans $\leftrightarrow$ English.

Mobile speech to speech translation systems for military applications, specifically the force protection domain were investigated in the DARPA funded Babylon and the already mentioned TransTac project. The system "Speechalator" that was developed at Carnegie Mellon University for the Babylon project is described in Waibel et al. (2003a) and Waibel et al. (2003b). The "Speechalator" used an interlingua-based machine translation component.

Realizations of TransTac systems are for example described in Hsiao et al. (2006) and the already mentioned Bach et al. (2007) (Carnegie Mellon University System) and also at `www.iraqcomm.com` (SRI System). Some specifics of BBN's TransTac effort were published in Saleem et al. (2007). Saleem et al. (2007) investigate various techniques to improve the machine translation performance by data normalization, additional knowledge sources and morphological analysis of Iraqi-Arabic. IBM's effort in the TransTac project is based on the MASTOR (Multilingual Automatic Speech to Speech Translator) system usually running on a laptop, but it was also ported to a handheld device in Zhou et al. (2003) and Zhou et al. (2004). The system on the handheld device uses a statistical natural language understanding and generation component, which parses the input and produces target language output.

Schultz and Black (2006) identify the main issues in porting a speech to speech translation system to a new language pair from a speech recognition and speech synthesis standpoint.

Chapter 4

Low Cost Language Portability

4.1 Introduction

This chapter will discuss methods to effectively generate bilingual training data to be used to support additional language pairs. Bilingual corpora are the training data for statistical machine translation models. They are specifically needed to train the translation models and phrase tables. Phrase pairs are extracted from the bilingual data and are combined to form the translation hypothesis during decoding as outlined in chapter 2. Unfortunately, bilingual corpora are only available for a small number of language pairs and, therefore, have to be produced for new language pairs.

Pivot languages were already introduced in chapter 2 (Reichert and Waibel, 2004; Babych et al., 2007) and can somewhat ameliorate this situation. Instead of translating from the source to the target language directly, the source language is first translated to a pivot language which is then translated to the target language. For each new language it is then only necessary to have a bilingual corpus with the chosen pivot language. The most commonly proposed pivot language is English, as it is part of many bilingual corpora.

The standard approach to generate bilingual training data is to manually translate a given monolingual corpus in one of the languages to the other language. If no monolingual corpus is available in either language, it would be necessary to create one or translate a corpus from a third language. In the experiments here, it is generally assumed that a monolingual corpus is already available for one of the languages. This is also the usual case as one of the involved languages is often English or another major language with easy access to large monolingual corpora.

4.1.1 Human Translators

In order to get a correct translation of the corpora, human translators have to be hired. Websites like Proz (`www.proz.com`) and Translatorsbase (`www.translatorsbase.com`) offer a convenient way to collect quotes and hire individual translators and translation agencies for a large variety of languages[1]. The best results are usually achieved if the translator is native in the target language. It is only necessary to understand the source language and non-native speakers can correctly comprehend the concepts. It is, however, much harder for non-native speakers to generate a translation in the target language that is exactly like an utterance a native speaker would use in the respective situation, particularly concerning word choice.

Translators are usually paid per source language word. Prices per word can range between 0.01 USD and about 0.25 USD, depending on the type and size of the project, the involved languages and the availability of translators for the languages. The general price level in the respective country plays an important role as well. Table 4.1 shows some of the lowest prices offered for BTEC and other tourism phrase/dialog corpora of at least 100,000 words for selected languages (all translations from English). A general rule of thumb is that a professional human translator is able to translate about 3,000-4,000 words per day, which is also consistent with the experiences in the InterACT laboratory. The bilingual English/Iraqi-Arabic data provided to the members

Language	Price per word
Indonesian	0.01 USD
Modern Standard Arabic	0.02 USD
Vietnamese	0.03 USD
Chinese	0.03 USD
Japanese	0.03 USD
Russian	0.03 USD
Korean	0.04 USD
Thai	0.05 USD
Malay	0.08 USD

Table 4.1: Selected prices for human translations

of the TransTac project contains about 6 million English words. This means, the estimated effort to produce this data added up to over 7 years (for 1 translator) at a cost of over 120,000 USD.

[1] As of August 2008, Proz offered to post translation jobs between 454 languages and dialects

4.1.2 Scenario

In the standard and most common situation the problem is as follows: A monolingual corpus is available and the goal is to have this corpus translated by human translators to produce a sentence-aligned bilingual corpus. Time and/or cost constraints do not allow for the full corpus to be translated. The main concern are cost constraints, as multiple translators can be hired to work in parallel to match time constraints.

Most corpora in their original state contain quite a large number of partial repetitions and duplicate sentences. It is possible that some of these repetitions will be beneficial, but it can be assumed that not all of them are really necessary, and very repetitive sentences can be eliminated. Table 4.2 shows some parts of the BTEC corpus. A large number of sentences contain various repetitions and share phrases.

	...
130:	**I'd like** to make a hotel reservation.
131:	Do you have a room for tonight?
132:	How long do we stay here?
133:	**I'd like a** shave, please.
134:	**I'd like a** haircut.
	...
173:	**Another** one, please.
174:	**May I have another** glass of water?
175:	**May I have another** fork?
176:	I'll show you to your room.
	...
227:	Overseas **operator, please.**
228:	This is Mr. Sato in room one two three four.
229:	I'd like to call Tokyo, Japan.
230:	Miki Hayakawa.
231:	**Operator, please.**
	...

Table 4.2: Examples for repetitions in the BTEC corpus

Given the assumption that these repetitions are not beneficial for the translation performance, the task is to eliminate these sentences. The specific task is to sort the sentences based on their estimated importance. The top n sentences are then selected and sent to the human translators. The goal is that the returned bilingual (sub-)corpus will offer the best possible

translation performance. Table 4.3 shows an overview of this general scenario.

Start situation	Complete monolingual corpus is available
End situation	Bilingual (sub-)corpus is available
Objective function	Translation performance of final system
Constraint	Cost/Time

Table 4.3: General scenario

It is not always clear at the start of development how much money will be available for the translations. The money situation could also change, and it could be possible to translate additional sentences later. This could depend on the final translation performance. If the translation performance does not meet the intended goal, funds might be made available to achieve a better performance. As this will frequently be the case, the goal is to sort all the sentences in the corpus in a way that the top n sentences are always approximately optimal. For example, the top 1000 sentences should be the same, regardless if 1000, 5000 or 10,000 sentences can be translated.

Evaluation Method The overall goal is to have the maximum translation performance of the final translation system for all potential test sentences. As usual, the translation performance is measured with an unseen test set and it is assumed that this test set is generic enough to estimate the translation performance on any test set.

It will be necessary to measure the translation performance of the optimized and non-optimized data at various corpora sizes. It is definitely possible that an approach might give good results for very small sizes, while another approach might be better suited for larger sized corpora.

4.2 Related Work

4.2.1 Active Learning in Natural Language Processing

In general, this research can be regarded as an example of active learning. This means the machine learning algorithm does not just passively train on the available training data, but plays an active role in selecting the best training data (Cohn et al., 1996). Active learning, as a standard method in machine learning, has been applied to a variety of problems in speech and

language processing. Examples are parsing (Hwa, 2004), automatic speech recognition (Kamm and Meyer, 2002; Zhang and Rudnicky, 2006) and dialog systems (Hakkani-Tür et al., 2006).

Query by Committee and Query by Uncertainty Two common generic approaches to active learning are *Query by Committee* (QBC) and *Query by Uncertainty* (QBU). QBU is described in Thrun and Moeller (1992). In this method, a model is trained based on the already labeled data and it is applied to the unlabeled data. The sample for which the trained model has the highest "uncertainty" will be the next to be labeled.

QBC was introduced in Seung et al. (1992). Here, multiple different models are trained based on the already labeled data (e.g. by sampling), and all are applied to the unlabeled data. The data sample for which this committee of models has the most disagreement is chosen as the next sample to be labeled.

These approaches have been successfully applied to Part-of-Speech (POS) and named entity tagging and generally gave comparable performance (Haertel et al., 2008).

However, the methods are computationally very complex, as new models have to be trained and applied after each newly labeled data sample. This means the techniques can only be used if the training is fast or can be done incrementally. It has to also be possible to apply the trained models very quickly.

A batch operation does not seem easily feasible. If the models are not updated, similar samples would be chosen multiple times in a row. In a POS tagging task there could be high uncertainty for a certain phrase like "would like to buy" which might lead to the fact that then multiple sentences with this phrase ("I would like to buy a car", "We would like to buy a house",...) are chosen at once. The necessity to re-train the models also makes it difficult to organize the human labelers. They would have to use an integrated tool that selects the next sample based on the previous input.

This problem, and the overall training effort in machine translation, do not allow easy application of these techniques to the problem of low cost language portability. Re-translating the untranslated sentences over and over again to find the next sentence to give to the human translator would just be impossible in practical applications. For this reason, the techniques in section 4.3.1 do not consider the translations, but base the selection only on features of the source language sentences. This can be interpreted as QBU on the source side, as previously unseen words and n-grams are used to estimate the importance of a sentence. The method proposed in section 4.3.2

can also be regarded as a variant of QBU. Here the uncertainty of word to word translations are estimated based on their probability development.

Elicitation Corpora For machine translation specifically, some related work has been done for the transfer approach based on grammars. During the elicitation process, the collection of data is controlled depending on which grammatical features and specialties a language might contain. A basic question asked could be if a language distinguishes singular and plural nouns. If it does, the elicitation corpus will contain examples for this feature, otherwise those examples will be skipped (Probst and Levin, 2002; Probst and Lavie, 2004; Alvarez et al., 2006). The result will be "a resource dense with the right features, those for which the target language makes distinctions" as stated in Clark et al. (2008).

The selection mechanism for active learning is almost exclusively dependent on the respective problem. However, besides Eck et al. (2005a) and Eck et al. (2005b) no other publications related to producing bilingual training data for statistical machine translation are currently available.

Translation Model Adaptation It is important to note the difference between this approach and approaches to translation model adaptation, as presented in Hildebrand et al. (2005) or Lü et al. (2007). In these cases, sentences for the training corpus are selected from a larger corpus based on the test sentences and certain similarity measures. The goal is then to improve the translation performance compared to a non-adapted translation system.

Simple sub-sampling techniques are also based on the actual test data and the goal there is to limit the memory requirement and increase the speed in order to translate a known test set.

In the approach presented here, it is assumed that the test data is unknown at selection time, so the intention is to get the best possible translation system for every potential test set.

4.2.2 Web Crawling for Bilingual Corpora

An alternative to generating bilingual corpora using human translators is the automatic crawling for bilingual corpora on the World Wide Web. Various websites are offered in multiple languages and could be a source for parallel bilingual corpora. This was tested in the "Surprise Language Exercise" sponsored by DARPA in 2003 for the languages pairs Hindi → English and a dry-run for Cebuano → English (Oard, 2003; Lavie et al., 2003). Web crawling

has multiple problems that prevent it from being used in the intended scenarios here. The domain of bilingual web data is mainly news, a mismatch to the intended dialog domains in tourism, medicine and force protection. It is also doubtful that web crawling can gather large corpora for minority languages. Most websites are only available in the major languages to reach the largest possible audience.

Web data can, however, be a good source of monolingual data for a variety of languages.

4.2.3 Increasing Translator Productivity

On the side of the human translator, software is available that has the same goal: to minimize the actual manual work necessary by the human translator. A large number of tools is available here under the term *Computer-assisted translation* (CAT). Commonly used CAT programs are Trados, developed by SDL International (`www.trados.com`), and Transit XV, developed by Star Group (`www.star-group.net`). Various alternate programs are also available. The software supports the human translator with user dictionaries, terminology managers and general project management.

The related feature to the issue discussed here is the *Translation Memory* component. This component saves earlier translations for text segments and matches them to new segments. If the tool finds the same sentence again, it will offer to reuse the translation, thus saving translation time. If the match is only partial, the translator will have to check and likely correct the proposed translation. These tools will really only be useful for duplicate sentences or those differing in just one word. However, removing duplicate sentences before a monolingual corpus is translated is trivial and generally done in this application. This would be different in other, more traditional, translation jobs where mainly complete documents are translated. It is also doubtful if the translators using these tools would lower their price according to their increased productivity. Therefore, these tools would not actually lower the cost of the machine translation system.

4.2.4 Summary Related Work

Various related approaches are available and were discussed in the previous sections. The general idea to actively select training samples is well-known as active learning and has been applied to various tasks in speech and language processing. The problem with the standard approaches QBU and QBC is the computational complexity which does not allow to directly apply them to the production of bilingual data for statistical machine translation.

The concept of elicitation has been applied to transfer-based machine translation. The goal is also to limit the necessary amount of data to be labeled by selecting sentences based on language features. Active learning approaches have not been applied to the task of producing a bilingual corpora for statistical machine translation and this is the contribution of the methods proposed in the following sections.

Approaches to web crawling are trying to completely eliminate the human translation cost by directly locating parallel bilingual corpora on websites. This is available and could be very promising in the future, but the current domain and language coverage is too limited.

Available CAT tools allow the human translators to be more productive by eliminating the necessity to re-translate duplicate sentences and identifying sentences that only differ in one word. The human translator will still have the goal to provide all translations to his clients, but the tools allow him to lower the time spent on these sentences.

For the problem discussed here it would not be necessary to have the full corpus translated. If a sentence is identified as not containing additional information it can just not be translated. In turn the approaches to improve the production of the bilingual data for statistical machine translation could eventually help the human translator. Once he has translated the most important sentences the translation of other sentences could be predicted that the user can edit. This is similar to the methods proposed in Barrachina et al. (2008). Here the software also suggests translations, but the sentences are not sorted in an optimized order.

4.3 Sentence Sorting

This section will introduce various methods to sort the sentences of a monolingual corpus. After sorting, the top n sentences can be given to a human translator to produce a bilingual corpus.

The *static* sentence sorting approaches in the following section will not consider the returned translation but sort the sentences just based on features of the source sentences.

The *dynamic* sentence sorting approaches in section 4.3.2 will consider the translations and base the order of the remaining sentences on the returned translations.

4.3.1 Static Sentence Sorting

4.3.1.1 Coverage Based Approaches

It was mentioned that the optimization goal is the translation performance of the final system on any sentence, which is estimated based on a test set. It is well-known that the translation performance of a statistical machine translation system is heavily correlated with the quality of the word alignment, the word to word lexicon and the phrase table. Also, the ability of the language model to distinguish good and bad translation candidates in the target language plays an important role.

The intention is now to estimate the impact of each sentence that could potentially be translated. This is accomplished by certain features that model these factors on the sentence level. This will allow this approach to select approximately optimal subsets of sentences.

Vocabulary Coverage The most basic feature and goal for a bilingual training corpus is full vocabulary coverage. For each word, a word alignment can only be calculated if at least one sentence containing this word is available. It has to be assumed that each word that occurs in the monolingual corpus can potentially occur in the test set, and at least a word to word probability for a lexicon is necessary to translate it. Therefore, the first goal is to cover every word in the vocabulary at least once. It will initially be assumed that each sentence has the same translation cost. This task can then be formulated as follows:
Find the smallest number of sentences that cover the full vocabulary.

This problem is exactly analog to the well-known SET-COVER problem (Karp, 1972).

Theorem 1 *Given a universe U, and a family S of subsets of U, a cover is a subfamily $C \subseteq S$ of sets whose union is U. In the SET-COVER decision problem, the input is a pair (U, S) and an integer k; the question is whether there is a set covering of size k or less (k refers to the number of sets). In the set covering optimization problem, the input is a pair (U, S), and the task is to find a set covering which uses the fewest sets.*

The decision version of SET-COVER is NP complete, and the optimization version of SET-COVER is NP hard.

A trivial reduction of the stated problem to SET-COVER proves that the stated problem is also NP hard.

A simple greedy algorithm for SET-COVER at each stage chooses the set that contains the highest number of uncovered elements.

It can be shown that this algorithm achieves an approximation ratio of:

$$A(s) = \sum_{k=1}^{s} \frac{1}{k}$$

with s being the size of the largest set.

Additional inapproximability results also show that this greedy algorithm is essentially the best-possible polynomial time approximation algorithm for SET-COVER (Lund and Yannakakis, 1994; Raz and Safra, 1997; Feige, 1998; Alon et al., 2006). These results can be directly applied to the sentence selection situation, and an algorithm for optimal vocabulary coverage can be formulated: *Choose the sentence in each step that contains the highest number of words not yet covered.*

The following example illustrates that this algorithm is not optimal here. Given a corpus with 5 sentences:

1.	A B C D a b c d
2.	E F e f
3.	G g
4.	A B C D E F G
5.	a b c d e f g

Table 4.4: Example corpus

The greedy algorithm will choose the sentences in the given order until choosing the third sentence. At this point the whole vocabulary is covered. The optimal solution is, however, to choose sentences 4 and 5 which gives an approximation ratio of $\frac{3}{2}$ in this case. In the general case, an example constructed this way achieves an approximation ratio of $log_2(s)/2$, with s being the number of words in the longest constructed sentence.
Generally the greedy algorithm to sort sentences can be formulated:

Algorithm 1 Sentence sorting

```
Create empty sorted list
repeat
  For all sentences that are not in the sorted list
  Find sentence with highest score
  Add sentence with highest score to sorted list
until all sentences sorted
```

with the score for each sentence defined as:

$$score(\text{sentence}) = \sharp(\text{unseen words})$$

N-gram Coverage A high performance statistical machine translation system cannot, however, only rely on word to word translations, but greatly benefits from phrase translation pairs. Even closely related languages cannot simply be translated word for word, and it is necessary to include phrase translations as well. Phrases can cover local reorderings and grammatical differences. They generally provide a foundation for a high quality translation.

For this reason, a second goal is to cover longer n-grams in addition to words. It is unclear which lengths of n-grams should be considered and different possibilities will be investigated in the experiments in section 4.4.

The situation with n-grams is exactly equivalent to the situation with single words in the previous section, and the analog algorithm can be applied with the adjusted sentence score. The parameter j will be used to indicate which lengths of n-grams are considered. In all experiments j was chosen as 1, 2 or 3. (This score includes the previous score if j is set to 1):

$$score_{N,0,j}(\text{sentence}) = \sum_{n=1}^{j} \sharp(\text{unseen n-grams})$$

Translation Cost Up to now, it was assumed that the cost to translate each sentence is constant and the goal was to find the smallest number of sentences. This is not the case, as human translators are paid per word and common corpora contain sentences of greatly varying lengths.

To adjust the calculated scores for each sentence, the scores are divided by the sentence length in words.

$$score_{N,1,j}(\text{sentence}) = \frac{\sum_{n=1}^{j} \sharp(\text{unseen n-grams})}{\text{sentence length}}$$

This score can be interpreted as the *unseen n-grams gained per translated word* for the respective sentence.

N-gram Importance Another simplification made in the beginning was that all words and n-grams have the same value. It does not matter for the previous terms if a sentence only contains very rare words or if the words are very frequent. However, it will be far more important to cover the most frequent words and n-grams compared to infrequent ones. The most relevant basis available to estimate the probability for an n-gram to occur in the test sentences is the monolingual source corpus. Therefore, the estimation of the

n-gram importance will be based on this data[2]. The obvious choice is to use the frequency for each n-gram as its importance:

$$score_{F,1,j}(\text{sentence}) = \frac{\sum_{n=1}^{j} \sharp(\text{frequency of unseen n-grams})}{\text{sentence length}}$$

Cover N-grams Multiple Times In order to get good word alignments and reliable phrase translation pairs, it might not be sufficient to cover each word and n-gram only once. It could be beneficial, especially for very ambiguous words, to collect multiple translations of that word in various sentences and contexts. Just based on the source language side, it is not possible to estimate how ambiguous the translations of a certain word might potentially be, but a certain goal can be set to cover each word and n-gram k times instead of once. Here the seen/unseen status of an n-gram is no longer sufficient in the scoring term so a *value* is introduced:

$$score_{MULT,1,j}(\text{sentence}) = \frac{\sum_{n=1}^{j} \sum_{\text{n-gram in sentence}} value(\text{n-gram})}{\text{sentence length}}$$

In the previous sections, the value function would only return two values. It would return the n-gram frequency in the frequency-based approach or 1 in the coverage-based approaches for an unseen n-gram. For previously seen n-grams, 0 would be returned. Here the value function will return positive values if the respective n-gram was seen less than k times, and 0 if it was seen k times or more. The initial value for unseen n-grams will either be 1, as in the coverage-based approach, or the n-gram frequency, as in the frequency-based approach. Various options exist for the intermediate values:

- Constant value: value stays at the same level until n-gram is covered k times
- Linear decrease: value decreases linearly until n-gram is covered k times
- Quadratic decrease: quadratic value decrease until n-gram is covered k times
- Exponential decrease: exponential value decrease until n-gram is covered k times.

[2]Other monolingual corpora or web search engines could be used to estimate the importance, but this was not investigated.

The steeper the decrease, the more importance is placed on unseen n-grams. In the constant case, any n-gram seen less than k times will have the same value as an unseen n-gram. This is most likely not beneficial, as an unseen n-gram will probably be more important in this situation.

However, the results showed that this score could not outperform the approaches that only intend to cover the words and n-grams once. This result was true regardless of the chosen setting.

The reason is that these additional enforcements of multiple translations are not generally necessary. Frequent words will automatically be covered multiple times in the previous scores, as they will most likely occur again in sentences that are selected later (without giving added score to the sentence). Rare words will most likely be covered only once. However, this can still result in a good alignment quality for the rare word if the other words in the sentence are reliably aligned. It is also the case that rare words are generally less ambiguous than frequent words. This was shown in Twilley et al. (1994) for the English language and simplifies the alignment task.

However, ambiguous words and phrases do exist and the goal of the scores introduced in section 4.3.2 is to be adaptive to each individual word. For an ambiguous word, multiple instances should be selected which will not be necessary for an unambiguous word. Section 4.3.2 will introduce a method that incorporates this based on the already received translations.

Prefer Shorter Sentence The algorithms for training translation models in statistical machine translation usually work better (and faster) on shorter sentences. The number of possible word alignments grows exponentially with the length of the sentence. This advantage is particularly obvious in the case of a one word sentence where the alignment is trivial. However, a one word sentence does not contain any context or phrase information and would not be preferable.

In order to prefer shorter sentences, the scores were also divided by the square of the sentence length. The index i in the scores will indicate this setting:

$$score_{N,i,j}(\text{sentence}) = \frac{\sum_{n=1}^{j} \sharp(\text{frequency of unseen n-grams})}{\text{sentence length}^i}$$

4.3.1.2 Information Retrieval

An alternative approach to scoring sentences is based on information retrieval techniques, particularly similarity measures. The TF-IDF similarity measure (Salton and Buckley, 1988) was chosen here, but this idea can easily be applied to any other similarity measure, such as Okapi. Other experiments, however, showed that the similarity measures do not behave very differently in various natural language processing applications and usually give nearly equivalent results (Hildebrand, 2005). This is not surprising, as they are based on comparable concepts.

TF-IDF is a similarity measure widely used in information retrieval. To calculate TF-IDF scores, each document D is represented by a vector $(w_1, w_2, \ldots, w_m)$ with the size of the overall vocabulary m. The entry w_k is calculated as: $w_k = tf_k * log(idf_k)$ with:

- tf_k is the term frequency (TF) of the k-th word in the vocabulary in the document D i.e. the number of occurrences.

- idf_k is the inverse document frequency (IDF) of the k-th term, given as

$$idf_k = \frac{\sharp\text{documents}}{\sharp\text{documents containing } k\text{-th term}}$$

The similarity between two documents is now defined as the cosine of the angle between the two vectors. The minimum angle is 0 degrees which means that the two vectors point in exactly the same direction, thus having the same vocabulary and relative frequencies, resulting in the maximum score of 1. The maximum angle is 90 degrees, which means that the two documents have completely disjunct vocabularies, resulting in the minimum score of 0.

Sentence Scoring with TF-IDF The approach is to use TF-IDF to find the most *different* sentence compared to the already selected sentences and give this one the highest importance. This means the sentence with the lowest TF-IDF score (compared to the already selected sentences), is selected to be translated next.

The following example will illustrate this approach. The very first sentence has to be randomly selected because there is nothing to compare the available sentences against in the first step.

The randomly selected sentence could be:

1.	*Where is the hotel?*

In the next step, the TF-IDF score for every still available sentence compared to this sentence is calculated. Sentences that do not have a single word in common with this sentence will get the lowest possible TF-IDF score of 0, and one of those will again be randomly selected, for example:

1.	Where is the hotel?
2.	*I had soup for dinner.*

At some point there will be no more sentences left that only contain unseen words, so every sentence will get a positive TF-IDF score. The lowest TF-IDF score will then be assigned to sentences that have the lowest number of already seen words and the highest document frequency for these words. A selected sentence in this example could be:

1.	Where is the hotel?
2.	I had soup for dinner.
3.	*This is fine.*

This sentence only shares the word "is" with the already sorted sentences. The word "is" most likely has a very high document frequency, thus a low IDF score. This leads to an overall low score for this particular sentence. A sentence like "We ate dinner at a restaurant." will get a higher score because the shared word "dinner" is certainly less frequent than "is" and will get a higher IDF score. The TF score in this example would be the same, so it can be ignored. In the next iteration, the TF score for "is" in the sorted sentences will be higher, which in turn lowers the chances to select another sentence with "is".

In summary, this scoring scheme will ensure that, at the beginning, new and unseen words are covered. It will also assign a higher score to more frequent words later, which is the same behavior as the scoring schemes presented in the previous sections.

A more information-retrieval centered motivation for the TF-IDF method is: Always select the sentence with the topic that is *furthest away* from the topic(s) of the sentences already sorted. This will make sure that all possible topics that are in the training data and may come up in the test data are covered.

Generalizing TF-IDF for N-grams TF-IDF can be easily generalized to n-grams by using every n-gram as an entry in the document vectors (instead of only using words). This will assign a higher score to phrases, as argued in the previous sections with the coverage based approaches. In the experimental section, it was investigated using words ($score_{TF/IDF,1}$) and bigrams ($score_{TF/IDF,2}$).

4.3.1.3 Additional Computational Complexity

This section will further discuss questions concerning the complexity of the proposed approaches. A different formulation of the sentence sorting problem is the following: Each sentence has a specific score, based on the number of previously unseen n-grams it contains or its TF-IDF score. A cost is attached to each sentence based on its length in number of words. It is important to realize here that the sentence scores might potentially change based on the other selected sentences. This problem, in a formal definition, can be reduced to the KNAPSACK problem (Kellerer et al., 2004).

The KNAPSACK problem is defined as: Given a set of items, each with a cost and a value, determine the number of each item to include in a collection so that the total cost is less than a given limit and the total value is as large as possible. The 0-1 KNAPSACK problem specifically states that each item can either be chosen or not chosen and is formally specified as:

Theorem 2 *Given n kinds of items, 1 through n. Each item j has a value p_j and a weight w_j. The maximum weight that can be carried in the bag is c.*

$$\begin{aligned} \textit{Maximize} &: \sum_{j=1}^{n} p_j x_j \\ \textit{subject to} &: \sum_{j=1}^{n} w_j x_j \leq c, x_j = 0 \textit{ or } 1, j = 1 \ldots n \end{aligned}$$

The decision version of 0-1 KNAPSACK is NP complete, and the optimization version is NP hard.

The sentence selection (SENTENCE SELECT) problem can be formalized as follows: Given n sentences, 1 through n. Each sentence j has a score p_j and a translation cost w_j. The maximum translation cost that can be afforded is c.

$$\begin{aligned} \textit{Maximize} &: \sum_{j=1}^{n} p_j x_j \\ \textit{subject to} &: \sum_{j=1}^{n} w_j x_j \leq c, x_j = 0 \text{ or } 1, j = 1 \ldots n \end{aligned}$$

The only significant difference is that the values for the individual sentences change based on the other selected sentences. This obviously makes the problem harder, but the following reduction also proves that the sentence selection problem is NP hard.

Any 0-1 KNAPSACK problem has to be reduced to a sentence selection problem. This means, given a number of weights and values, sentences have to be designed that represent the same scores and costs. For the vocabulary coverage approach, this is accomplished as follows: For each new item, a sentence is generated with as many new and different words as the items value. The rest of the sentence is filled with the last word until the intended weight is reached. Once this new sentence selection problem is solved, it also gives a solution to the 0-1 KNAPSACK problem: The items corresponding to the selected sentences have to be chosen.

There are a number of instances where the reduction is not as obvious as stated:

- It is not possible to generate a sentence corresponding to an item value of 0. Those items can, however, be automatically eliminated as they do not generate any benefit if chosen.
- Item values for KNAPSACK problems are usually natural numbers. For rational values and weights, the values, weights, and the overall maximum cost can be multiplied by appropriate factors to generate natural numbers.
- Sentences generated according to this method will always have a cost that is at least as high as their score. An appropriate factor can again be introduced to match this condition with the values and weights of the items.

For more information about KNAPSACK problems in general, please consult Pisinger (1995) and Kellerer et al. (2004). Table 4.5 shows an example of the reduction of 0-1 KNAPSACK to SENTENCE SELECT. The reduction can be realized with only polynomial effort, which proves that the sentence selection problems are also NP hard and NP complete respectively.

0-1 KNAPSACK	
Value	Cost
2	4
3	3
2	2
1	5

$\xrightarrow{Reduction}$

SENTENCE SELECT		
Sentence	unseen	Cost
$A_{1,1}\ A_{1,2}\ A_{1,2}\ A_{1,2}$	2	4
$A_{2,1}\ A_{2,2}\ A_{2,3}$	3	3
$A_{3,1}\ A_{3,2}$	2	2
$A_{4,1}\ A_{4,1}\ A_{4,1}\ A_{4,1}\ A_{4,1}$	1	5

Table 4.5: Example for a reduction of 0-1 KNAPSACK to SENTENCE SELECT

4.3.2 Dynamic Sentence Sorting

4.3.2.1 Basic Idea and Approach

One problem with the previous sorting approaches is that they do not consider the actual translation that is received for a certain sentence. Therefore, the sorted order of the sentences is the same for all target languages.

This is reasonable for purely coverage based approaches, but if words and n-grams are supposed to be translated multiple times to get better alignments, the target language has significant impact. This is comparable to the elicitation corpora for grammar-based approaches. In these cases, the features of the target language influence the selection of sentences to be translated. (see section 4.2 and Probst and Levin (2002); Probst and Lavie (2004); Alvarez et al. (2006)). If a word has a clear one-to-one translation, not too many training examples will be required to get quality alignments. On the other hand, if the translation of a word is very ambiguous, more training examples might be necessary. At least all translation variants have to be covered once in the bilingual corpus.

The general ambiguity of a word in the source language plays an important role here, but this also depends on the target language. If each meaning of an ambiguous word translates to a different word in the target language, far more training examples will be necessary than if a single word in the target language exists that has exactly the same ambiguities. In this case, only one training example could be adequate.

4.3.2.2 Probability Development

The two charts in figure 4.1 demonstrate the development of IBM1 probabilities with an increasing number of sentences. The first chart shows the probabilities for the English word "bank" and various Spanish translations (the top translation candidates). The IBM1 training data contains 10,000 lines of BTEC data that did *not* contain the word "bank" and then 1, 2, ... 100 lines

with the word “bank” are added. So the overall training data at each position consists of 10,001, 10,002, ...10,100 lines. In the second chart of figure 4.1, the same experiment was done with the word “and”. The 10,000 lines of BTEC were added to get the most common words in the data already well aligned, therefore, the alignments for the chosen words are less affected by other words.

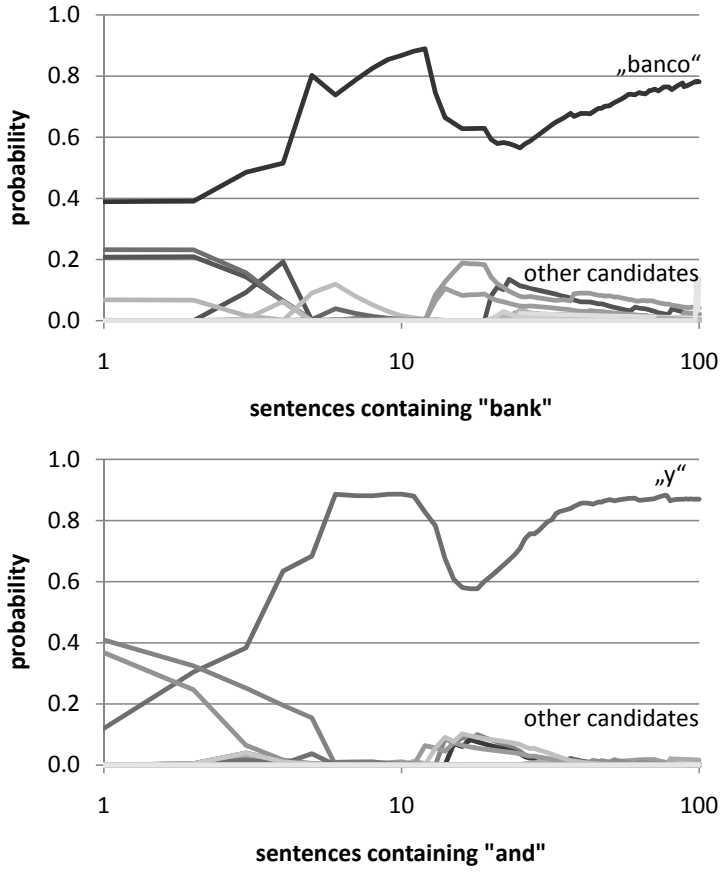


Figure 4.1: IBM1 probabilities English → Spanish for words “bank” and “and”

In both cases, it is clear that early on one translation (“y” for “and” and “banco” for “bank”) gets most of the probability mass. After about 10-20 sentences containing the respective word, the probabilities no longer change considerably, and all other candidates stay at marginal probabilities. There is also no change in the candidate with the highest translation probability for “bank” over the course of the experiment. The word “and” behaves slightly differently, but the candidate “y” also gets the highest translation probability

quite early and stays at a high probability.

A very different situation is shown in the charts in figure 4.2. Here the same experiments were done again with the words “put” and “nice”.

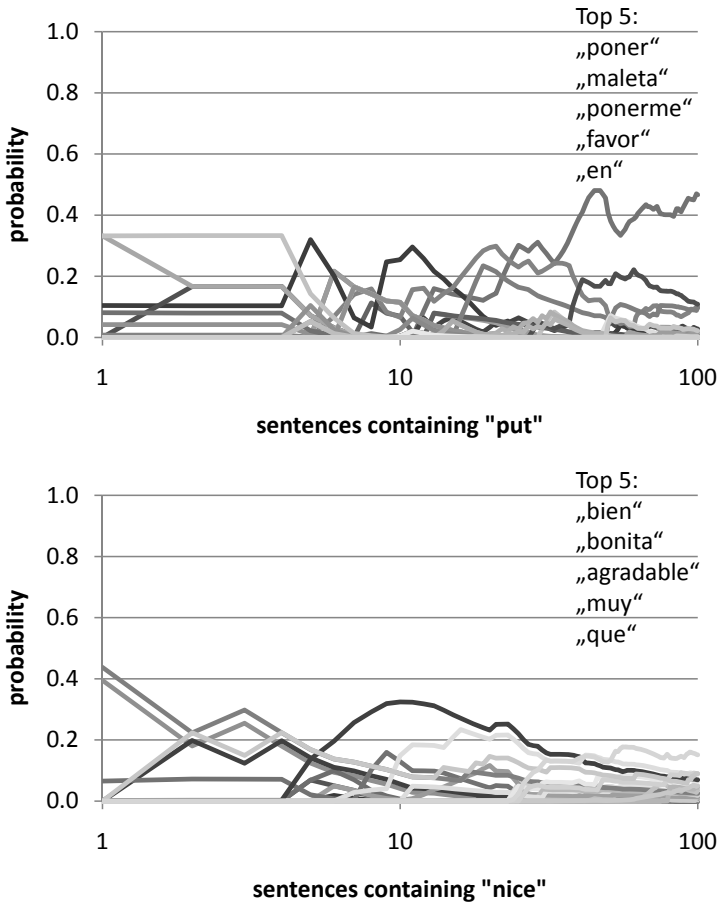


Figure 4.2: IBM1 probabilities English → Spanish for words “put” and “nice”

The word “put” is very ambiguous in English. While “nice” is not as ambiguous, it can be used in many different contexts which could lead to varied translations. The IBM1 probabilities consequently show a very different behavior. The probabilities are generally very unstable, and a translation candidate seems to receive more probability mass only at the very end.

The fundamental idea of this approach is to check the development of the word alignment probabilities over more and more sentences to decide if a certain word might benefit from additional training examples or if that is unlikely. In the four examples shown, it is clear that for “bank” and “and” the probabilities seem to remain stable early on, and additional sentences

are not necessary. For "put" and "nice" the probabilities are never as stable and it could be concluded that adding more sentences would improve the stability.

4.3.2.3 Scoring of Sentences

To calculate sentence scores based on the probability development, two probability vectors, $\boldsymbol{p_k}$ and $\boldsymbol{p_{k+1}}$ are introduced. $\boldsymbol{p_k(s)}$ contains the IBM1 probabilities $p(s|t)$ for the source word s and all words t in the target vocabulary with the selected corpus containing the source word s k times. $\boldsymbol{p_{k+1}}$ is the analog vector when the corpus contains source word s $k+1$ times.

Using the cosine distance, these vectors can be compared. If there is no change between $\boldsymbol{p_k}$ and $\boldsymbol{p_{k+1}}$, the cosine distance will be 0. If there are differences, the cosine distance can have values up to 1.

This allows the definition of a *cos-dist* (cosine distance) score for each word at each sorting stage. The rest of the scores are analog to the previously defined static sentence scores. This can be done in a coverage based version $score_{DN,i,j}$ and a frequency based version $score_{DF,i,j}$:

$$
\begin{aligned}
score_{DN,i,j}(\text{sentence}) &= \frac{\sum_{n=1}^{j} \sharp(\text{unseen n-grams})}{\text{sentence length}^i} \\
&+ \frac{\sum_{word \in sentence} \text{cos-dist(word)}}{\text{sentence length}^i} \\
score_{DF,i,j}(\text{sentence}) &= \frac{\sum_{n=1}^{j} (\text{frequency of unseen n-grams})}{\text{sentence length}^i} \\
&+ \frac{\sum_{word \in sentence} \text{frequency of word} * \text{cos-dist(word)}}{\text{sentence length}^i}
\end{aligned}
$$

If a word is still unseen, the cosine distance function is not calculated.

A problem pointed out in section 4.2 regarding the QBU and QBC techniques is the increased computational complexity. The IBM1 probabilities have to be re-calculated after each sentence is sorted. Afterward, the cosine distance for each word has to be recalculated as well. However, some simplifications and optimizations are possible here. First, the IBM1 probabilities can be recalculated after a batch of sentences has been sorted. To avoid the

over-generation of specific words, the cosine distance should be reset to 0 after that word is chosen within a batch. After a batch is finished, the cosine distances will also be recalculated. A second simplification is the recalculation of the cosine distances of only those words that were actually present in the added sentences. This is not exact, as the cosine distances for other words might also change based on the IBM1 probability shift during recalculation. The experiments showed, however, that this has little influence.

It would theoretically be possible to generalize this approach to n-grams exactly analog to the words. However, the computational complexity will become so high that it will not be useful for practical purposes.

4.3.3 Summary Sentence Sorting

This section will give a summary of the scores for sentence sorting that were introduced and defined in the preceding sections before presenting and discussing the experimental results in the next section.

Static Sentence Sorting The scores for static sentence sorting consider only the source sentences.

$score_{N,i,j}$: Coverage based score with the goal of covering each n-gram once.

$$score_{N,i,j}(\text{sentence}) = \frac{\sum_{n=1}^{j} \sharp(\text{unseen n-grams})}{\text{sentence length}^i}$$

$score_{F,i,j}$: Frequency based score with the goal of covering each n-gram once. N-grams are weighted by frequency.

$$score_{F,i,j}(\text{sentence}) = \frac{\sum_{n=1}^{j} \sharp(\text{frequency of unseen n-grams})}{\text{sentence length}^i}$$

$score_{TF/IDF,1}$ and $score_{TF/IDF,2}$: Topic based score. Select sentence next that has the lowest TF/IDF score against the already sorted sentences. $score_{TF/IDF,2}$ includes bigrams in the TF/IDF calculation.

Dynamic Sentence Sorting The scores for dynamic sentence sorting also take the translations into account. The value for words is adjusted based on the IBM1 probability development.

$score_{DN,i,j}$ Coverage based score with dynamic score adjustment for words.

$$\begin{aligned} score_{DN,i,j}(\text{sentence}) &= \frac{\sum_{n=1}^{j} \sharp(\text{unseen n-grams})}{\text{sentence length}^i} \\ &+ \frac{\sum_{word \in sentence} \text{cos-dist(word)}}{\text{sentence length}^i} \end{aligned}$$

$score_{DF,i,j}$ Frequency based score with dynamic score adjustment for words.

$$\begin{aligned} score_{DF,i,j}(\text{sentence}) &= \frac{\sum_{n=1}^{j} (\text{frequency of unseen n-grams})}{\text{sentence length}^i} \\ &+ \frac{\sum_{word \in sentence} \text{frequency of word} * \text{cos-dist(word)}}{\text{sentence length}^i} \end{aligned}$$

4.4 Experimental Results

The main part of the experiments in this chapter were done using an English → Spanish translation system. An additional validation experiment was later done on Thai → English.

Machine Translation System The applied statistical machine translation system for these experiments uses the PESA online phrase extraction algorithm based on IBM1 lexicon probabilities (Vogel, 2003, 2005; Eck et al., 2006). All language models are trigram language models with Kneser-Ney-discounting built with the SRI-Toolkit (Stolcke, 2002).

4.4.1 Experiment English → Spanish

Test and Training Data All translations in these experiments were done translating English to Spanish. The training data here consisted of 123,416 lines with 903,525 words on the English side, and 852,362 words on the Spanish side of tourism phrase corpora. The test data consisted of 500 lines of dialogs from the medical domain.

Bilingual Training Data			Monolingual Training Data	
	English	Spanish		Spanish
Lines	123,416	123,416	Lines	123,416
Words	903,525	852,362	Words	852,362
Translation Models	PESA online			
Language Model	SRI 3-gram			
Test Data	500 lines, medical dialogs			
Baseline Score	0.141 (BLEU), 4.19 (NIST)			

Table 4.6: Experimental setup English → Spanish

Baseline Coverage For unsorted data, the token coverage for unigrams, bigrams and trigrams on the English part of this data shows a relatively linear behavior, as illustrated in figure 4.3. Overall the English data contains 12,578 distinct unigrams, 98,397 distinct bigrams and 208,452 distinct trigrams. The type coverage situation will be very different, as only a small number of frequent unigrams make up the large majority of the data.

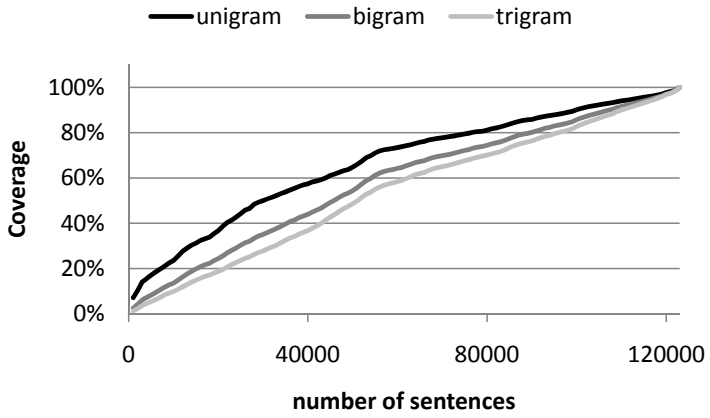


Figure 4.3: Unigram, bigram and trigram coverage for unsorted data

Baseline Translation Scores It was necessary for these experiments to have different baseline systems in order to compare the performance for different training data sizes. For each system at each step the language and translation models have to be re-trained. The baseline system that uses all available training data achieved a BLEU score of 0.141 [0.131; 0.152] and a NIST score of 4.19 [4.03; 4.35] (95% confidence intervals). For the baseline systems, that do not use all available training data, the sentences were used in the original random order of the training corpus. Translation systems trained on this (smaller) data give the BLEU and NIST scores shown in figures 4.4 and 4.5 respectively.

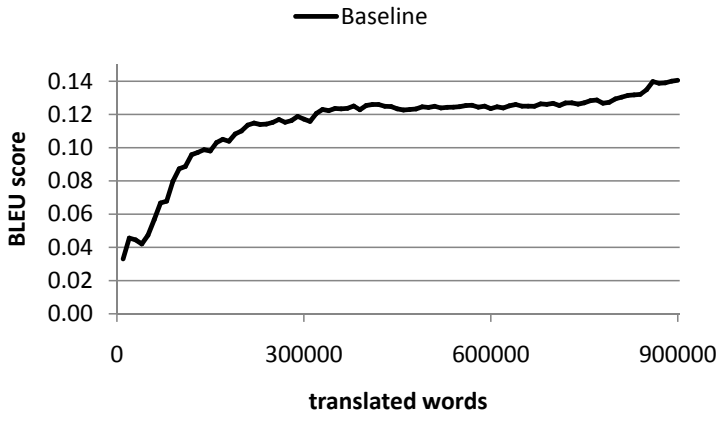


Figure 4.4: Baseline BLEU scores - data in original order

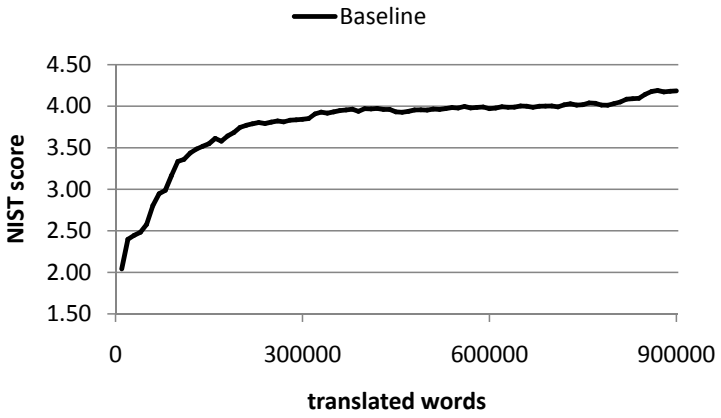


Figure 4.5: Baseline NIST scores - data in original order

The diagrams clearly show a rather steep increase of the scores until the translation of approximately 400,000 words; the scores increase only slightly until they reach the final score for the system using all available training data. The results also show that the behavior of the NIST scores is very similar to the BLEU scores. There were some small differences in the later experiments, but the overall results were the same. For this reason, the NIST scores are omitted here. The publications Eck et al. (2005a) and Eck et al. (2005b) show NIST score results.

4.4.2 Static Sentence Sorting

4.4.2.1 Optimized Coverage

The vocabulary coverage and n-gram coverage based sentence scores $score_{N,0,1}$, $score_{N,0,2}$, and $score_{N,0,3}$ focus exclusively on the coverage. Therefore, it was first examined how much coverage the sorted sentences would achieve. In these scores, each sentence is implicitly assumed to have the same cost, and the goal is to find the smallest number of sentences covering all considered n-grams. The coverage for $score_{N,0,1}$ is shown in figure 4.6. The graph also shows the number of translated words in the sentences. This value is rather well correlated to the number of sentences, which indicates that the sentence lengths are not varying significantly.

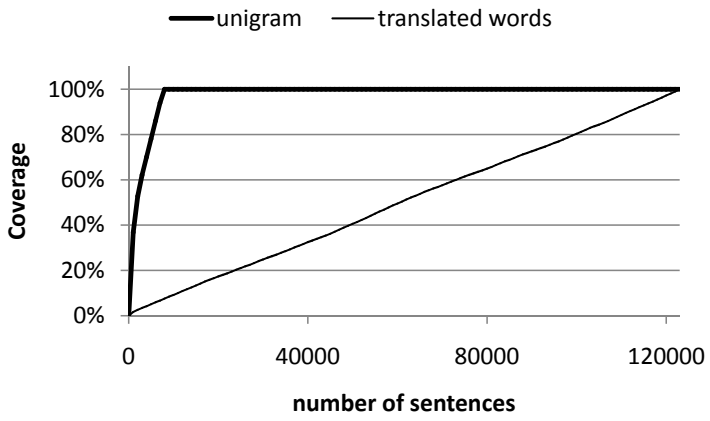


Figure 4.6: Optimization according to $score_{N,0,1}$

Full vocabulary coverage in this case is reached at sentence 7,763 after translating 66,582 words. This means after only 6.3% of the sentences (7.4% of the words), full vocabulary coverage is achieved.

Figures 4.7 and 4.8 show the coverages for $score_{N,0,2}$ and $score_{N,0,3}$. These sentence scores also consider unseen bigrams ($score_{N,0,2}$) and unseen bigrams and trigrams ($score_{N,0,3}$). In a similar picture, as in figure 4.6, the coverages are growing very fast, and only a small number of sentences is needed to achieve full coverage. For $score_{N,0,2}$, all unigrams and bigrams are covered after 39,234 sentences (31.8%) and 315,443 translated words (34.9%) while it takes 62,096 sentences (50.3%) and 484,855 translated words (53.7%) to cover all unigrams, bigrams and trigrams when sorting according to $score_{N,0,3}$.

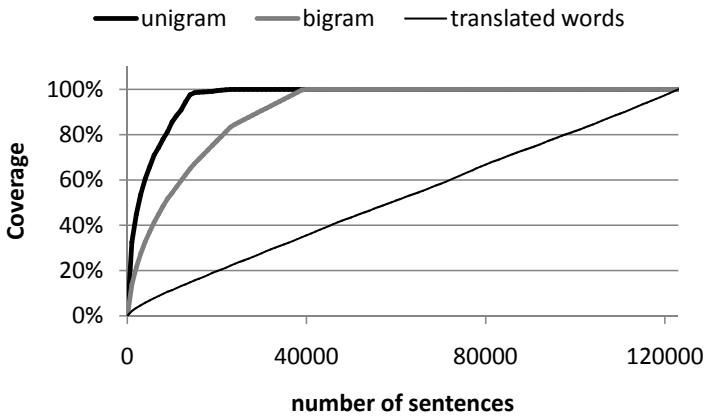


Figure 4.7: Optimization according to $score_{N,0,2}$

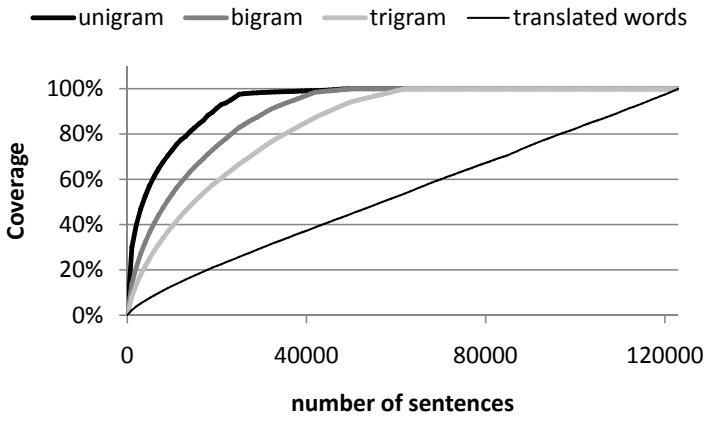


Figure 4.8: Optimization according to $score_{N,0,3}$

4.4.2.2 Translation Results

Results for $score_{N,0,j}$ Figure 4.9 illustrates the BLEU scores for systems where the sentences were sorted according to $score_{N,0,j}$.

If the optimization only uses the number of previously unseen unigrams to rank a sentence, the systems receive significantly higher BLEU scores than the baseline for very small amounts of training data. However, the steep increase stops very soon, and the systems fall below the baseline. The translation scores recover again at about 500,000 translated words. The reason for this pattern is most likely that the optimization achieves a much better coverage for the smaller amounts of training data, but after a while, the baseline system reaches a similar coverage of the testing data and has a

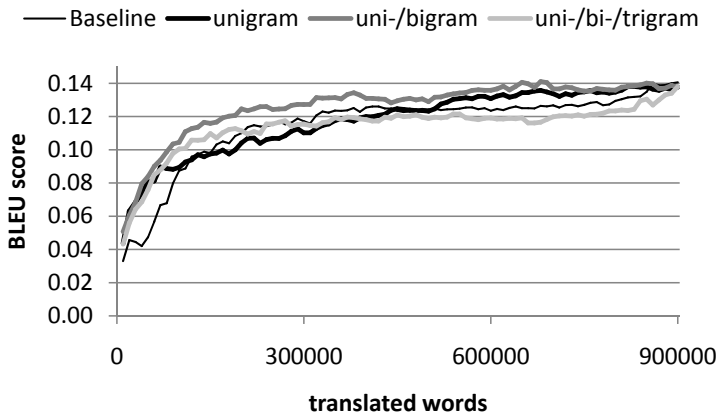


Figure 4.9: Results for data sorted according to $score_{N,0,j}$

more meaningful language model with more realistic frequencies.

These problems are clearly fixed by incorporating the bigrams into the optimization process. The BLEU scores no longer fall below the BLEU scores of the baseline systems, but stay consistently higher. Incorporating trigrams gives, again, lower BLEU scores.

The optimization based on uni- and bigrams reaches a BLEU score of 0.130 at 320,000 and a BLEU score of 0.135 at 570,000 translated words. A BLEU score of 0.130 is only about 8% worse and a BLEU score of 0.135 is only about 4% worse than the baseline BLEU score of 0.141 (achieved when training on the whole training data). Scores of 0.135 are already in the confidence interval of the baseline system, so it is highly probable that these systems are not significantly worse than the best baseline system.

Results for $score_{N,1,j}$ The difference between $score_{N,0,j}$ and $score_{N,1,j}$ is the incorporation of the length of a sentence. The number of unseen n-grams is divided by the number of words in this sentence to get the score for the sentence. This more closely models the cost factor of a translation that is paid per word. These results are shown in figure 4.10.

A comparison with figure 4.9 shows that the BLEU scores for the sorting of the sentences according to $score_{N,1,j}$ are even better than for the term $score_{N,0,j}$. The optimization based on unigrams shows a very similar behavior to $score_{N,0,j}$, with the same lower BLEU scores after translating about 200,000 words and a BLEU score recovery toward the end. The optimizations based on uni- and bigrams and uni-, bi- and trigrams are clearly improved compared to $score_{N,0,j}$. There are no significant differences between the optimization based on uni- plus bigrams and the optimization incorporating

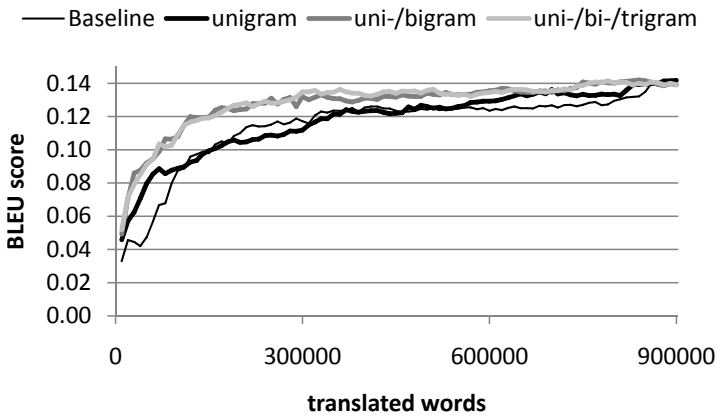


Figure 4.10: Results for data sorted according to $score_{N,1,j}$

trigrams, too. In this case, a BLEU score of 0.13 was already reached at 250,000 translated words.

Results for $score_{N,2,j}$ As explained in section 4.3.1.1, it was also tried to prefer shorter sentences with $score_{N,2,j}$ by dividing the number of unseen n-grams by the square of the number of words in the respective sentence. The reason is that shorter sentences might be easier to align, as fewer possible word alignments have to be considered. Figure 4.11 shows that this did not further improve the results achieved using term $score_{N,1,j}$, but gave lower BLEU scores.

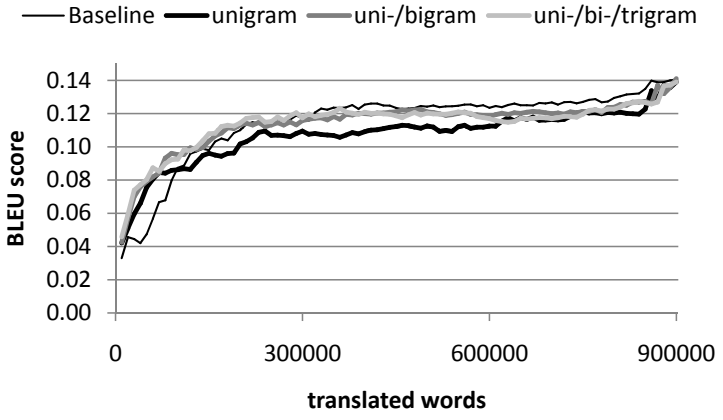


Figure 4.11: Results for data sorted according to $score_{N,2,j}$

Results for $score_{TF/IDF}$ Figure 4.12 shows the BLEU scores for the optimization based on TF-IDF for unigrams and uni-/bigrams. In this case, the original TF-IDF (based only on unigrams) slightly outperforms the TF-IDF based on uni- and bigrams, but neither approach shows better results than the previously introduced sentence scores with the expection of very small amounts of data.

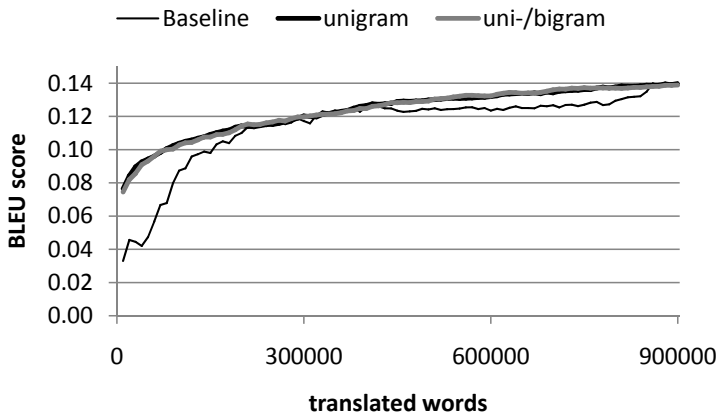


Figure 4.12: Results for data sorted according to $score_{TF/IDF,1}$ and $score_{TF/IDF,2}$

Results for $score_{F,0,j}$ Figure 4.13 illustrates the BLEU scores for systems where the sentences were sorted according to $score_{F,0,j}$.

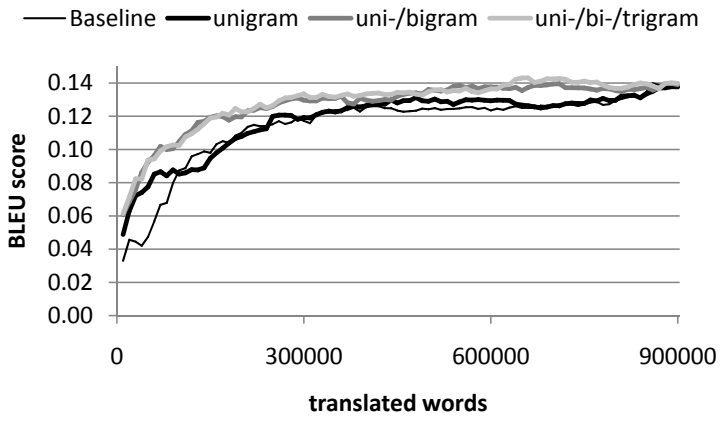


Figure 4.13: Results for data sorted according to $score_{F,0,j}$

If the optimization only uses the frequency sum of previously unseen unigrams to rank sentences, the systems receive significantly higher BLEU scores than the baseline for very small amounts of training data. However, the steep increase stops very early and the systems fall slightly below the baseline, recover toward the end, and finish on the same BLEU scores. These problems are clearly fixed by incorporating the bi- and trigrams into the optimization process. The scores no longer fall beyond the scores of the baseline systems but stay consistently higher. The systems optimized on uni- and bigrams ($score_{F,0,2}$) are not significantly different from the systems for uni-/bi- and trigrams ($score_{F,0,3}$) but show a very similar performance.

Results for $score_{F,1,j}$ The difference between the term $score_{F,0,j}$ and $score_{F,1,j}$ is the incorporation of the length of a sentence. The frequency sum of the unseen n-grams is divided by the number of words in the respective sentence to get the score for the sentence. Figure 4.14 illustrates the associated BLEU scores.

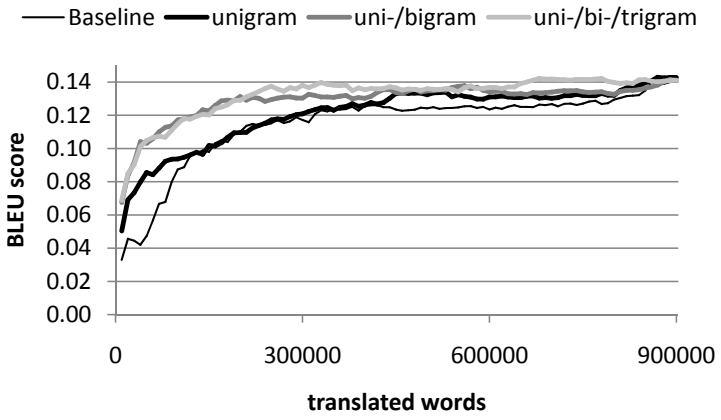


Figure 4.14: Results for data sorted according to $score_{F,1,j}$

A very similar behavior can be observed for the unigram based score. The results show an improvement for the optimizations based on uni- and bigrams and uni-/bi- and trigrams compared to $score_{F,0,j}$. There are no significant differences between the BLEU scores for those two optimizations. The performance is very similar, with only slight advantages for the optimization based on uni-/bi- and trigrams ($score_{F,1,3}$).

Results for $score_{F,2,j}$ As explained in section 4.3.1.1 and analog to $score_{N,2,j}$ it was also tried to prefer shorter sentences in term $score_{F,2,j}$ by dividing the frequency sum of the unseen n-grams by the square of the number of words in the respective sentence. Diagram 4.15 illustrates those BLEU scores.

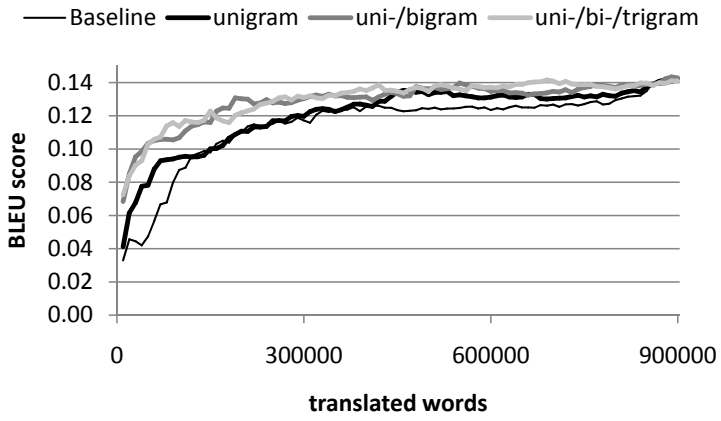


Figure 4.15: Results for data sorted according to $score_{F,2,j}$

4.4.3 Sentence Score Comparison

The diagrams in the preceding sections nicely showed the improvements over the baseline, but a score comparison was not easily possible. As the results are most interesting and different for smaller amounts of training data table 4.7 lists the results for data sizes of 10,000, 20,000, 50,000, 100,000 and 200,000 translated words.

BLEU scores shown in bold are the best scores in each column. The general results remain. It is always valuable to consider unigrams and bigrams, but trigrams either do not help or merely improve the scores insignificantly. The sentence scores should be divided by the square of the sentence length. The sentence score based on TF/IDF performs best for the smallest number of words, but does not improve for the later iterations and falls behind.

	Score for number of translated words				
	10k	20k	50k	100k	200k
Baseline	0.033	0.046	0.047	0.087	0.110
Coverage based scores					
$score_{N,0,1}$	0.044	0.063	0.081	0.090	0.104
$score_{N,0,2}$	0.051	0.061	0.084	0.105	0.125
$score_{N,0,3}$	0.043	0.056	0.076	0.101	0.111
$score_{N,1,1}$	0.046	0.057	0.080	0.088	0.104
$score_{N,1,2}$	0.050	0.071	0.092	0.108	0.124
$score_{N,1,3}$	0.052	0.071	0.091	0.109	0.127
$score_{N,2,1}$	0.042	0.051	0.075	0.086	0.102
$score_{N,2,2}$	0.043	0.057	0.079	0.095	0.113
$score_{N,2,3}$	0.046	0.059	0.080	0.093	0.114
Scores including frequency					
$score_{F,0,1}$	0.049	0.063	0.078	0.085	0.108
$score_{F,0,2}$	0.061	0.069	0.092	0.105	0.120
$score_{F,0,3}$	0.061	0.071	0.094	0.102	0.123
$score_{F,1,1}$	0.050	0.070	0.086	0.094	0.110
$score_{F,1,2}$	0.068	0.084	0.103	**0.117**	**0.131**
$score_{F,1,3}$	0.069	**0.085**	**0.105**	0.115	0.130
$score_{F,2,1}$	0.041	0.062	0.078	0.095	0.111
$score_{F,2,2}$	0.068	**0.085**	0.103	0.107	0.130
$score_{F,2,3}$	0.072	0.084	0.103	0.114	0.122
TF/IDF based scores					
$score_{TF/IDF,1}$	**0.077**	0.084	0.094	0.104	0.114
$score_{TF/IDF,1}$	0.075	0.082	0.093	0.103	0.114

Table 4.7: BLEU score comparison for static sentence scores.

4.4.4 Dynamic Sentence Sorting

The results for the dynamic sentence sorting are compared to the best performing static scores in table 4.8. Batch sizes of 100 and 1000 were used.

	Score for number of translated words				
	10k	20k	50k	100k	200k
Baseline	0.033	0.046	0.047	0.087	0.110
Coverage based scores					
$score_{N,1,1}$	0.046	0.057	0.080	0.088	0.104
$score_{N,1,2}$	0.050	0.071	0.092	0.108	0.124
$score_{N,1,3}$	0.052	0.071	0.091	0.109	0.127
$score_{DN}$ with batch size 100					
$score_{DN,1,1}$	0.047	0.059	0.082	0.091	0.107
$score_{DN,1,2}$	0.052	0.071	0.093	0.107	0.126
$score_{DN,1,3}$	0.055	0.070	0.091	0.109	0.126
$score_{DN}$ with batch size 1000					
$score_{DN,1,1}$	0.046	0.058	0.083	0.090	0.108
$score_{DN,1,2}$	0.050	0.073	0.095	0.106	0.122
$score_{DN,1,3}$	0.052	0.070	0.090	0.107	0.127
Scores including frequency					
$score_{F,1,1}$	0.050	0.070	0.086	0.094	0.110
$score_{F,1,2}$	0.068	0.084	0.103	0.117	**0.131**
$score_{F,1,3}$	**0.069**	**0.085**	**0.105**	0.115	0.130
$score_{DF}$ with batch size 100					
$score_{DF,1,1}$	0.052	0.072	0.087	0.097	0.111
$score_{DF,1,2}$	**0.069**	0.083	0.104	**0.119**	0.129
$score_{DF,1,3}$	**0.069**	0.084	**0.105**	0.117	0.129
$score_{DF}$ with batch size 1000					
$score_{DF,1,1}$	0.050	0.070	0.086	0.094	0.110
$score_{DF,1,2}$	**0.069**	0.083	0.102	0.117	**0.131**
$score_{DF,1,3}$	**0.069**	**0.085**	**0.105**	0.115	0.130

Table 4.8: BLEU score comparison for dynamic scores.

The resulting BLEU scores are very close to the original scores and only show small, insignificant improvements occasionally. The biggest increases can be seen for the scores that only consider the unigrams.

4.4.5 Experiment Thai → English

In order to validate the positive results, the sorting according to ($score_{N,1,2}$) was also applied to the task of translating Thai to English in the medical domain.

Bilingual Training Data			Monolingual Training Data	
	Thai	English		English
Lines	59,191	59,191	Lines	59,191
Words	422,692	457,736	Words	457,736
Translation Models	PESA online			
Language Model	SRI 3-gram			
Test Data	496 lines, medical dialogs			
Baseline Score	0.294 (BLEU), 5.99 (NIST)			

Table 4.9: Experimental setup Thai → English

Machine Translation System The applied statistical machine translation system for these experiments uses the PESA online phrase extraction algorithm based on IBM1 lexicon probabilities (Vogel, 2003, 2005; Eck et al., 2006). The Language models are trigram language model with Kneser-Ney-discounting built with the SRI-Toolkit (Stolcke, 2002).

Test and Training Data The whole training corpus for these experiments had 59,191 sentences with 457,736 English words from the medical domain. The training data was also available in Thai with 422,692 words. The test data consisted of 496 lines, also taken from the medical domain.

Baseline Systems Table 4.10 shows different baseline scores for these systems. Even with more than 43,000 sentences - more than two-thirds of the whole data - the scores are still 28% (NIST) and 40% (BLEU) lower than if all training data is used.

♯ sentences	♯ English words	BLEU	NIST
38,000	306,231	0.172	4.30
43,000	345,773	0.176	4.29
59,191	457,736	0.294	5.99

Table 4.10: Baseline scores for Thai → English translations

Results The BLEU and NIST scores in table 4.11 clearly indicate that the sorted sentences achieve significantly better results than the baseline systems. The system trained on only 10,000 sentences clearly outperforms the NIST score of the baseline systems trained on 43,000 and 38,000 sentences and reaches an only slightly lower BLEU score. At 30,000 sentences, the BLEU score is only 7% lower than the highest score with the NIST score being only 2% lower. Figure 4.16 illustrates the results. This shows that, overall very similar results on a different task and language pair are possible.

♯ sentences	♯ English words	NIST	BLEU
5,000	43,040	4.11	0.104
10,000	82,997	4.84	0.169
20,000	187,595	5.79	0.263
30,000	319,405	5.86	0.274
40,000	395,374	5.92	0.280
59,191	457,736	5.99	0.294

Table 4.11: Scores for Thai → English translations after optimizations according to $score_{N,1,2}$

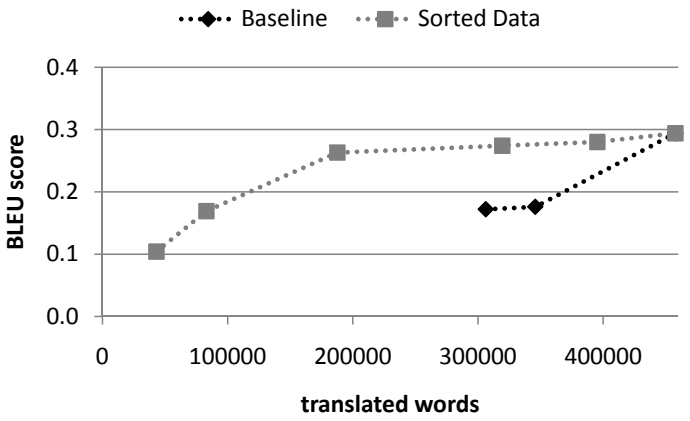


Figure 4.16: Results for Thai → English

4.5 Conclusions

For all practical applications, the fastest and comparably well-performing approach is to sort the sentences by the number of unseen uni- and bigrams divided by the sentence length. It is valuable to also consider the frequency, but the slightly improved results require a higher computational complexity, as all frequencies have to be determined. The improvements here are mainly insignificant.

The dynamically sorted approach is far more computationally intensive and relatively complicated to implement in a practical situation. It requires the source sentences to be sent in small batches to the translators. Not all translators would agree to that or may charge a higher fee. The added communication effort could also eliminate potential cost savings, and the improvements will generally not be worth the added effort. The experiment

Reference:	Su corazón late normalmente.
Baseline at 50k words:	Y tu heart est beating normally.
Best system at 50k words:	Su corazón latía normalmente.
Source sentence:	Your heart is beating normally.
Reference:	¿tengo herpes?
Baseline at 50k words:	podría darme herpes?
Best system at 50k words:	tengo herpes?
Source sentence:	I have herpes?
Reference:	Un poco, pero no mucho.
Baseline at 50k words:	Un poco excesivo, pero servirá mucho
Best system at 50k words:	Un poco, pero no mucho.
Source sentence:	A little bit, but not much.

Table 4.12: Example translations at 50,000 translated words

on English → Spanish showed that the sorted sentences can heavily improve the baseline scores. A BLEU score of 0.131 could be reached at 200,000 translated words which is less than 25% of the overall data. This means in turn that the human translation cost can be reduced by 75%, while a performance is reached that is within the confidence interval of the performance of a system trained on all available data.

It might be questionable wether very small data sizes have any practical application. However, the examples in table 4.12 show some nice improvements at the small level of 50,000 translated words (for English → Spanish).

On Thai → English the results are similar, but more data is necessary to reach the same levels of performance. In this case the savings are in the range of 50%.

These results certainly depend on language pair and specific language characteristics, but they will be heavily influenced by the type of data used as well.

Chapter 5

Models for Mobile Devices

5.1 Introduction

As discussed previously, a tourist, medical professional or warfighter will demand a lightweight and mobile translation system, and can not be expected to carry a heavy computer. So, the final speech to speech translation system should be able to run on a PDA, a cell phone or a similarly small device, such as a hand held game console. The main problems with running translation systems on such small devices are the lack of computing power and the memory requirements of the translation and language models.

The necessary computing power can be limited by using integer based (or fixed point) calculation as shown in Hsiao et al. (2006) and later in Zhang and Vogel (2007), but the memory requirements remain. The main memory-intensive parts are the translation and language models used during the decoding process. It is unknown which sentence will need to be translated, so sub-sampling is impossible.

For the language model a lossy hashing technique using a Bloom filter data structure was proposed in Talbot and Osborne (2007a,b), and this drastically lowers the memory requirements to a quite manageable size for standard n-gram language models. Standard n-gram language models are also generally smaller than the translation models. However, no solution has been proposed so far for the memory reduction of the translation model. Therefore, the focus here will be the translation model.

The translation model in a standard statistical machine translation system is usually a phrase table that contains triples of source and target words or phrases with a number of scores. The scores are weighted after parameter tuning and can be combined to form a translation probability or combined score (see chapter 2).

Start situation	Trained translation models are available that do not meet the memory constraints of the intended device.
End situation	Models that meet the memory constraints of the intended device
Objective function	Translation performance of final system

Table 5.1: Situation and goals in this chapter

Unfortunately, these phrase tables cannot be compressed using the Bloom filter technique. A Bloom filter can only store boolean membership information, but cannot store a number of translation candidates given a source phrase. Other clever compression or encoding techniques might be beneficial, but it is unlikely that enough memory can be saved by these measures and no techniques have been proposed yet.

The standard approach in this case is *pruning*, which means phrase pairs that are estimated to have little impact on translation quality are removed until the translation model fits the memory requirements.

It is impossible to pre-determine what the exact memory restrictions will be in every possible mobile device. This will certainly increase over time, but it is also very dependent on the actual device. Table 3.4 in chapter 3 lists a number of mobile devices and their memory specifications. This memory has to be shared with other applications so only a part of it will actually be available.

For this reason, the general intention of the presented algorithms is to find approaches that are independent of the specific size constraints and scale according to changing situations. The goal is to offer good performance at all possible and reasonable sizes (see table 5.1).

Online Phrase Pair Extraction An alternative approach to phrase extraction is *online* extraction from the bilingual training data instead of having a pre-extracted phrase table with millions of phrase pairs. Online extraction dynamically extracts phrase pairs necessary for the actual test sentence as proposed in Callison-Burch et al. (2005) and Zhang and Vogel (2005). These techniques usually improve the performance, as they can match arbitrarily long phrases. At the same time, they need more computing power compared to pre-extracting the phrase pairs, so they will most likely not be used for small devices. For this reason, only translation models consisting of pre-extracted phrase pairs will be discussed here (offline phrase tables).

5.2 Related Work

Threshold Pruning The standard approaches, which are usually applied in cases where the required space for a phrase table has to be constrained, are simple threshold pruning strategies. Standard threshold pruning limits the number of translation candidates or gives a minimum translation probability. Further candidates or candidates with a lower probability are removed. As this is a fairly simple approach, no publications are available that researched this specifically; however, it is directly available in the open source decoders Pharaoh (Koehn, 2004) and Moses (Koehn et al., 2007). Both pruning methods will serve as a baseline and will be further introduced in section 5.3.1.

Improvement via Pruning Conceptually related to this research are approaches that try to *improve* the translation quality by removing phrase pairs from a phrase table. Two methods were presented in Zettlemoyer and Moore (2007) and Johnson et al. (2007). Both publications identify incorrect and questionable phrase pairs that can be deleted and realize small to significant improvements in translation quality. They also try to identify redundant phrase pairs similar to the work presented here. The main difference to the approaches in this thesis is the goal of the algorithms and the assumption. In the algorithms and experiments here, the assumption is that the given phrase table is close to optimal and has to be pruned for the sole purpose of constraining the memory requirements. This assumes that every phrase pair can potentially add "value" under certain conditions, and the goal is to find phrase pairs where these conditions are very unlikely. This means a score improvement was not intended and could not be expected under these circumstances. This assumption is relatively naive given the current models, but the presented algorithms could still be used after the ideas shown in Zettlemoyer and Moore (2007) and Johnson et al. (2007) were applied to a given phrase table. On the other hand, Zettlemoyer and Moore (2007) and Johnson et al. (2007) do not have the goal to limit the memory requirements and cannot produce arbitrarily small phrase tables that still perform comparably well.

Bloom Filter Language Models For the language models, the papers Talbot and Osborne (2007a,b) use a Bloom filter data structure to efficiently store n-gram language models. A Bloom filter (Bloom, 1970) is a space-efficient probabilistic data structure, specifically a lossy hash, used to test whether an element is a member of a set. False positives are possible, while false negatives are not. If the overall number of elements to be stored is

known, the use of multiple concurrently applied hashing functions can lower the approximate expected loss. Both papers are mainly concerned with using very large language models on regular computers, but this approach can be applied to smaller language models on small devices as well.

5.3 Generating Small Translation Models

5.3.1 Threshold Pruning

As mentioned, threshold pruning is a well known and simple method to eliminate phrase pairs that will most likely not be needed in the decoding process. Probability threshold pruning and translation variety threshold pruning are both directly available in the Pharaoh (Koehn, 2004) and Moses (Koehn et al., 2007) decoders.

Probability Threshold Pruning A very simple way to prune phrase pairs from a translation model is to use a probability threshold and remove all pairs for which the translation probability is below the threshold. The reasoning for this is that it is very unlikely that a translation with a very low probability will be chosen over another translation candidate with a higher probability. This is, however, not impossible as other models, particularly the language model, might actually prefer the candidate with the low probability and boost the score enough so that it will be chosen in the final translation hypothesis.

Translation Variety Threshold Pruning Another way to prune phrase pairs is to impose a limit on the number of translation candidates for a certain phrase. This means the pruned translation model can only have equal or fewer possible translations for a given source phrase than the threshold. This is accomplished by sorting the phrase pairs for each source phrase according to their probability and eliminating low probability ones until the threshold is reached. This can also be interpreted as assigning a *rank* to each phrase pair based on the probability and eliminating all phrase pairs below a certain rank.

Phrase Pair Re-combination An additional threshold pruning method is a way to eliminate longer phrases by finding shorter phrase pairs that produce the same output with the same or a similar probability. The intention is not to eliminate phrases that are rarely or never used. Here it is possible that commonly used phrases are eliminated. However, these phrases are

not necessary, as the same output can be accomplished using shorter phrase pairs. For example, given a phrase pair for a translation system translating Spanish → English:

Necesito examinar su cabeza ⟶ I need to examine your head

This could be replaced by different combinations of shorter phrase pairs e.g.:

Option 1:
Necesito examinar ⟶ I need to examine
su cabeza ⟶ your head

Option 2:
Necesito ⟶ I need
examinar ⟶ to examine
su cabeza ⟶ your head

If these shorter phrase pairs are also in the translation model with a significant probability, it can be assumed that they will be put in the translation lattice as well.

Given a phrase pair translating $s_1\ s_2\ \ldots\ s_i$ to $t_1\ t_2\ \ldots t_j$ with the probability P.

$$s_1\ s_2\ \ldots s_i \longrightarrow t_1\ t_2\ \ldots t_j : P$$

Additional phrase pairs are now sought with the property that the concatenation of their source sides forms the original source side ($s_1\ s_2\ \ldots\ s_i$) and the concatenation of their target sides forms the original target side ($t_1\ t_2\ \ldots t_j$):

$$\begin{aligned}
s_{1,1}\ s_{1,2}\ \ldots s_{1,i_1} &\longrightarrow t_{1,1}\ t_{1,2}\ \ldots t_{1,j_1} : p_1 \\
s_{2,1}\ s_{2,2}\ \ldots s_{2,i_2} &\longrightarrow t_{2,1}\ t_{2,2}\ \ldots t_{2,j_2} : p_2 \\
&\ldots \\
s_{n,1}\ s_{n,2}\ \ldots s_{n,i_n} &\longrightarrow t_{n,1}\ t_{n,2}\ \ldots t_{n,j_2} : p_n
\end{aligned}$$

The overall probability to choose this group of phrase pairs as a concatenation is $p = \prod_{l=1}^{n} p_l$:

If $p = P$, then the combination of phrase pairs is equally likely to be chosen as the original phrase pair[1].

[1]Purely based on this probability. Additional features during decoding might prefer one option over the other even with the same phrase translation probabilities. It could be beneficial to prefer a candidate translation that uses longer phrase pairs.

A potential problem could be that a reordering step might rearrange the combination of the shorter phrases, while it would not change the word order of the longer one. It is also unlikely that it will be possible to find shorter phrases for which the combined probability exactly matches the probability of the longer phrase pair. Therefore a ratio r is defined as the maximum of the probability ratios for a given split:

$$r = \arg\max(\frac{p}{P}, \frac{P}{p})$$

A threshold for this r-ratio allows for the elimination of phrase pairs based on a maximum discrepancy in the probabilities.

If even more phrases need to be removed, it would be possible to further soften the conditions by only asking for a close match of the target or even source side using edit distance or other similarity measures. For longer phrases especially, small discrepancies could probably be neglected, but this was not investigated as the approach presented in the next section clearly outperformed this idea.

Computational Complexity One general problem with this method is that it is computationally complex to consider all possible source side splits. For a source side phrase with n words there are $n-1$ possible split positions between each of the words. There are two options for each of the $n-1$ split-positions: the phrase is split at this point or it is not split at this point. This leads to $2^{(n-1)}$ possibilities to split the source side. The trivial situation of no splits does not have to be investigated, so the overall number of splits to consider is $2^{(n-1)} - 1$, which has to be done for all phrase pairs.

5.3.2 Pruning via Usage Statistics

The last approach presented here uses a different idea inspired by the Optimal Brain Damage algorithm for neural networks as described in Cun et al. (1990).

5.3.2.1 Optimal Brain Damage for Neural Networks

The Optimal Brain Damage algorithm for neural networks computes a *saliency* for each connection weight in the neural network. The saliency is the estimated relevance for the performance of the network. In each pruning step, the connection weight with the smallest saliency is removed, the network is retrained and all saliencies are re-calculated. Once a network node has no more outgoing, edges the node can also be pruned.

Algorithm 2 Optimal Brain Damage

Train net into a minimum of the network
repeat
  Compute the saliency for each unit respectively
  Prune the element with the smallest saliency
  Retrain the net
until intended network size reached

Figure 5.1 illustrates the general concept of this algorithm.

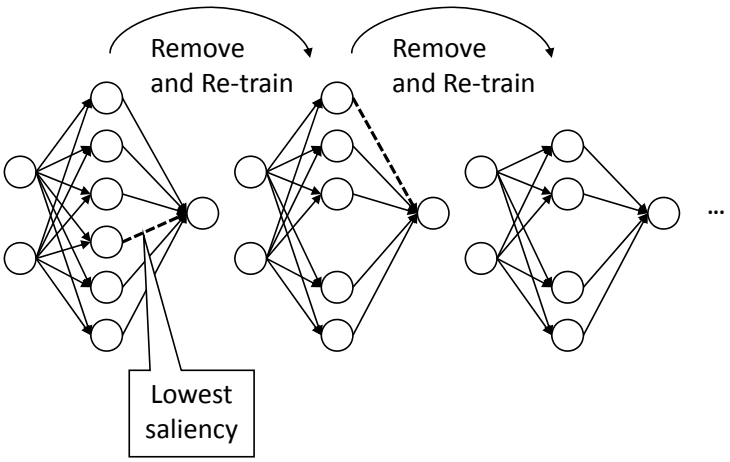


Figure 5.1: Optimal Brain Damage algorithm for Neural Networks

5.3.2.2 Transfer to Translation Models

Each phrase pair in the translation system can be analogously viewed as such a network connection weight. The only remaining question is how to calculate the relevance for the performance for each phrase pair. A simple approximation was already done in section 5.3.1 with the threshold pruning. Here the relevance was estimated using the phrase pair probability and the phrase pair rank (for phrase pairs with the same source side) as relevance indicators. However, these are not the only factors that influence the final selection of a phrase pair, and most of these factors are not established during the training and phrase extraction processes. The following two additional factors play a major role for the importance of a phrase pair.

Frequency of the Source Phrase It is clear that a phrase pair with a very common source phrase (e.g. "where is the hotel?") will be much

more important than a phrase pair where the source phrase occurs only very rarely (e.g. a very specific phrase: "the phone number is 555-274-6545"). A common phrase will most likely be used in more test sentences and will have a higher impact on the overall performance.

Actual Use of the Phrase Pair Even phrase pairs with very common source phrases might not be used for the final translation hypothesis. One reason could be a low probability for this phrase pair or other influencing factors that eliminate this particular phrase pair from the consideration. It is, for example, possible that it is part of a longer phrase pair that gets a higher probability so that the shorter phrase pair is not used. This is also influenced by the translation system, as one decoder might have a tendency to not choose the shorter phrase pair while another one might prefer it.

Overall, the following factors clearly influence the relevance of a phrase pair:

- Phrase pair probability
- Phrase pair ambiguity
- Frequency of source phrase
- Context of phrase usage

However, there are many other factors influencing the estimated importance of a phrase pair, and it is difficult to consider each factor separately. Consequently, the proposed idea does not use a combination of pre-established features to estimate the phrase pair importance. Instead, the idea is to just apply the translation system to a large amount of text, collect "real-life" usage statistics for each phrase pair and base the pruning decisions on these statistics.

5.3.2.3 Generic Pruning Algorithm

The generic pruning algorithm is then quite simple as illustrated in figure 5.2.

After training the models, a large amount of data is translated and usage statistics for the phrase pairs are collected. The pruning step is based on these usage statistics.

The pruning itself can be done in different ways. Phrase pairs can simply be continuously pruned based on the original usage statistics. A second option is to re-translate the text after each pruning step (removing one or a certain number of phrase pairs) and collect new usage statistics for the

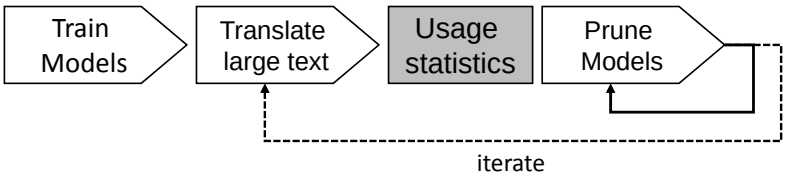


Figure 5.2: Generic pruning algorithm

remaining phrase pairs. The latter option is more computationally complex than the continuous pruning but might offer an improved pruning performance. Both variants are indicated by the arrows in figure 5.2.

5.3.2.4 Collecting Usage Statistics and Scoring Phrase Pairs

The remaining question is which usage statistics are actually relevant for a phrase pair. There are three possibilities concerning how a phrase pair can be used during the decoding of a specific sentence.

1. **Phrase Pair is *not* used** The first possibility is that a phrase pair is not considered at all. One reason that a phrase pair would not be considered at all during the translation of a sentence is that the source side of the phrase pair does not occur in the sentence so it would not be applied. Another possible reason is that its probability or its rank within the phrase pairs with this source phrase is too low to be considered, as it is outside of the decoding beam.

2. **Phrase Pair occurs in Lattice** The second possibility is that a phrase pair occurs (one or more times) in the translation lattice. This means the source phrase does occur in the sentence that is translated, and the beam factor did not eliminate that phrase pair from the consideration.

3. **Phrase Pair occurs in Hypothesis** The third possibility is that the phrase pair also occurs (one or more times) in the final translation hypothesis. This means that the phrase already occurred in the lattice, but it was also chosen to be part of the final translation, as the overall model-best path included this phrase pair.

Figure 5.3 shows an example for lattice occurrences and hypothesis occurrences. The phrase pair B occurs twice in the overall lattice here but does not occur in the bold path of the final translation hypothesis, while phrase pair A also occurs twice in the lattice, but also in the final translation path.

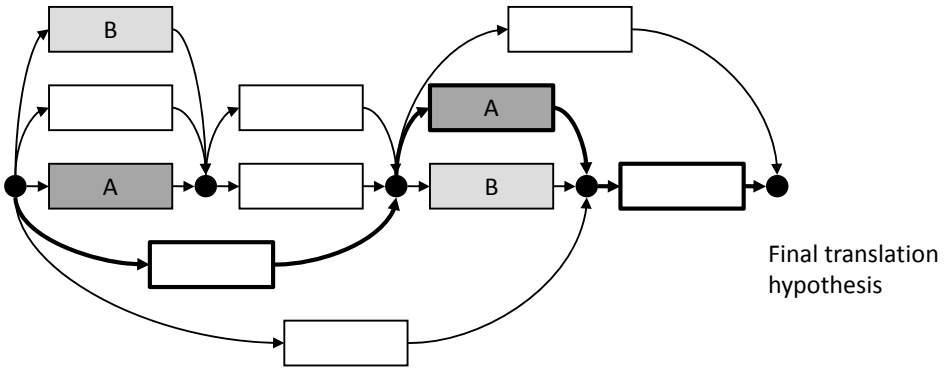


Figure 5.3: Lattice occurrences and hypothesis occurrence

In order to use these statistics for translation model pruning, an empirical term was first devised that showed significant improvements over other methods. Later, additional probability estimates were used that further improved these results. The findings were published in Eck et al. (2007b,a).

5.3.2.5 Empirical Scoring Term

Define the following two counts:

- c(phrase pair) = Count how often a phrase pair was added to the translation lattice.
- u(phrase pair) = Count how often a phrase pair was used in the final translation path.

For both counts it is reasonable to assume that a higher value will indicate a more important phrase pair. The source side of a phrase pair that is added to the lattice will occur frequently in the text, so the phrase pair will have a high influence on the actual translation performance. Furthermore, a phrase pair that is often found in the final translation path will be even more important, as it actually forms the translation.

It cannot, however, be assumed that a phrase pair that is never found in the lattices or final translation paths will never occur. This is a similar argument that is made for the discounting step in language modeling where probability mass has to be discounted from the seen events and assigned to unseen events. For that reason, a simple +1 discounting is introduced by adding 1 to each of the statistics per phrase pair. Also introduced is the log

function in order to limit the influence of the c count as the u count is likely to be more important.
Overall the empirical score function for a phrase pair is then:

$$score_{empirical}(\text{phrase pair}) = [\log(c(\text{phrase pair} + 1))] * [u(\text{phrase pair}) + 1]$$

Other combinations of these scores were attempted, but they did not outperform the results of this function.

As outlined earlier, the phrase pairs can now be sorted, and the phrase table can be pruned to the intended size.

In the following sections, a slightly different approach will be taken that tries to directly estimate the "usage" probability of a phrase pair based on its use in the model-best and metric-best paths.

5.3.2.6 Model-best Path Pruning

The fundamental idea of this approach to translation model pruning is to estimate how likely it is that a phrase pair will be used in the 1-best or model-best path of an N-best list. From a pure phrase pair perspective, each translation hypothesis in an N-best list can be viewed as a number of phrase pairs that were applied to the source sentence to generate this hypothesis. This is illustrated in figure 5.4.

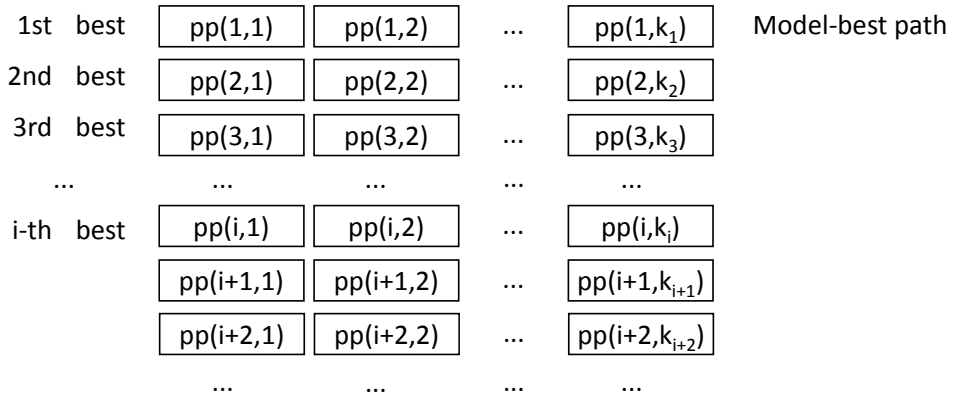


Figure 5.4: Phrase pairs of an N-best list

The i-best hypothesis is generated by the k_i phrase pairs $pp(i,1), \ldots pp(i,k_i)$. Please note that these phrase pairs do not have to be distinct. It is possible that one translation path is generated by duplicate phrase pairs if the original source sentence contains repetitions.

Neighboring hypotheses in the N-best list often share a number of phrase pairs and might differ in only one of them.

The path that will finally be chosen by the decoder is the path that gets the overall best score by all applied models (translation model, language model, distortion model etc.) which is the 1-best or model-best path. All other paths are disregarded.

To get the same 1-best entry and, therefore, the same final translation hypothesis for this particular sentence, it is only necessary to have the phrase pairs $pp(1,1), \ldots pp(1,k_1)$ in the translation model. All other phrase pairs that occur in the N-best list could be eliminated without changing the final (model-best) translation path for this particular source sentence. However, these phrase pairs might be used in the 1-best translation path of other sentences, so they cannot just be removed.

It is possible to estimate the probability that a phrase pair will be used in the 1-best translation path of any sentence by translating a large number of sentences and counting these occurrences. Based on a large number of translated sentences, the probability of a phrase pair occurring in the 1-best path can be estimated as:

$$P(\text{phrase pair in 1-best}) \approx \frac{\sharp\text{phrase pair in 1-best}}{\sharp\text{words in corpus}}$$

This is not divided by the number of sentences, as a phrase pair might be used multiple times within one sentence. Instead, it is divided by the number of theoretical chances it has to be applied which is the number of words in the corpus. This number is the same for all phrase pairs, so it can be ignored for these purposes.

This estimation is used as a means to assign a score to each phrase pair that should approximate the relative probability that it will be used in a 1-best translation.

$$score(\text{phrase pair}) = \sharp\text{phrase pair in 1-best}$$

The phrase pairs can then be sorted according to this score, and the top n phrase pairs can be selected for a smaller phrase translation model. However, this score does not discriminate very well, as only a relatively small number of phrase pairs occur in the 1-best translation even if a large number of sentences is translated.

Considering the 1-best to 10-best Translation Paths For this reason, the 2-best to 10-best translation paths were considered in addition to the 1-best path. To limit the influence of the 2-best to 10-best translation paths

these counts are divided by their index ($score_{A1}$), and in a second possibility the square of their index ($score_{A2}$). Overall, this assumes that a phrase pair that occurs frequently in the top 10 translation paths in the N-best list generally has a high probability of being in the model-best path. The following two scores are defined:

$$score_{A1}(\text{phrase pair}) = \sum_{n=1}^{10} \frac{\sharp(\text{phrase pair in i-best})}{i}$$

$$score_{A2}(\text{phrase pair}) = \sum_{n=1}^{10} \frac{\sharp(\text{phrase pair in i-best})}{i^2}$$

Please note that the number of times a phrase pair occurs in the lattice is no longer explicitly used in these and the following scores. However, the remaining phrase pairs that did not have an assigned score were sorted by the number of times they occurred in the lattices.

5.3.2.7 Metric-best Path Pruning

Each N-best list contains a path (or a number of paths) that is the best path according to a scoring metric. To find this metric-best path, a reference translation has to be available. Figure 5.5 illustrates this situation with the i-best path as the metric-best path. All paths above and below the i-best path have a lower (or possibly equal) score according to this metric.

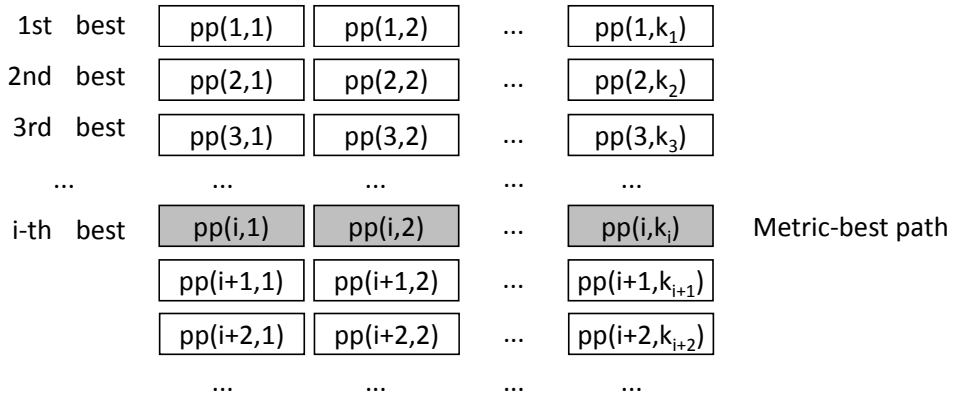


Figure 5.5: Phrase pairs of an N-best list

A problem with the previous pruning approach is that potentially good phrase pairs in the metric-best path might actually be removed and will

no longer be available for an unseen test sentence because the models did not value them enough during the collection of the pruning statistics.

For this reason, the second pruning approach considers the metric-best path. To avoid pruning of the phrase pairs in the metric-best path, the same statistics as in the previous section are applied, and two additional scores for a phrase pair are defined. Here the counts for each phrase pair in the top 10 paths according to a scoring metric are considered. In the experiments, the edit distance between a translation hypothesis and a reference translation was used as the scoring metric. The phrase pair scores $score_{B1}$ and $score_{B2}$ were defined analogously to section 5.3.2.6:

$$score_{B1}(\text{phrase pair}) = \sum_{n=1}^{10} \frac{\sharp\text{phrase pair in metric-i-best}}{i}$$

$$score_{B2}(\text{phrase pair}) = \sum_{n=1}^{10} \frac{\sharp\text{phrase pair in metric-i-best}}{i^2}$$

5.3.2.8 Pruning towards the Metric-best Path

The scores defined in the previous section do not actually enforce the metric-best path, but rather try to ensure that the phrase pairs within the metric-best path are not removed. In this section, an additional score is defined that intends to actively remove phrase pairs that eliminate the metric-best path from being chosen as the final translation hypothesis (i.e. being the model-best path).

In figure 5.5, the paths 1 to i-1 have a higher model score than the metric-best path at index i, while all paths with indices higher than i have a lower model score. If the paths 1 to i-1 could somehow be eliminated, the metric-best path would have the highest model score of all paths and become the translation hypothesis. Removing one phrase pair from each of the 1 to i-1 paths from the translation model would be enough to eliminate these paths from consideration.

After this pruning step is completed, the N-best list will contain new paths replacing these eliminated paths, but their model scores will be lower than the model score of the metric-best path at index i. If these new paths would achieve a higher model score, the decoder could have chosen them during the original decoding step[2].

[2]It is possible that early elimination of possible translation hypotheses during the decoding process could change that statement, but this behavior can still be considered very unlikely.

The following term intends to remove these unwanted phrase pairs:

$score_E$(phrase pair) = Number of times the phrase pair occurs in a path that has a higher model score than the metric-best path while *not* occurring in the metric-best path.

In this case, a high score would indicate the removal of a phrase pair. This score has to be used in combination with the other scores as it does not consider how often a phrase pair might actually occur in a metric-best path and should not be pruned.

The potential problem with this score is that it might not be possible to clearly classify phrase pairs into ones that will most likely occur in the metric-best path and ones that will most likely not occur in the metric-best path. A high number of phrase pairs could occur in both situations. Another problem with this approach is that it may not be possible to eliminate all of the paths with better model scores than the metric-best path. The remaining paths will still have better model scores, but possibly an even lower metric score. Here the pruning would be counter-productive.

5.4 Experimental Results

5.4.1 Experimental Setup

Machine Translation System All experiments were done with a state-of-the-art statistical machine translation system (Vogel, 2003; Eck et al., 2006). The system uses the phrase extraction method PESA described in Vogel (2005) and a 6-gram language model (Zhang and Vogel, 2006).

Training and Testing Data The training data for all experiments consisted of the Japanese-English BTEC corpus (Takezawa et al., 2002) with 162,318 lines of parallel text. The test set from the evaluation campaign of IWSLT 2004 (Akiba et al., 2004) was used as testing data. This data consists of 500 lines of tourism data. The first experiments were done translating Japanese → English. The results were later also validated in experiments translating English → Japanese. 16 English reference translations were available while there was only 1 reference on the Japanese side. The language model was trained on the target side of the bilingual training data.

Extracted Phrases Extracting phrases for n-grams up to a length of 10 (with low frequency thresholds) resulted in 4,684,044 phrase pairs

(273,459 distinct source phrases) for Japanese → English and 4,882,645 phrase pairs (453,201 distinct source phrases) for English → Japanese. The translation models with all phrase pairs achieved baseline scores of 0.5911 BLEU for Japanese → English and 0.1704 BLEU for English → Japanese with 95% confidence intervals of $[0.5713, 0.6109]$ and $[0.1659, 0.1752]$ respectively (see table 5.2).

Bilingual Training Data			Monolingual Training Data	
	English	Japanese		English
Lines	162,318	162,318	Lines	162,318
Words	1,003,785	1,188,106	Words	1,003,785
Translation Model	PESA phrase table			
Language Model	Suffix Array 6-gram			
Test Data	500 lines, tourism phrases (IWSLT 2004)			
Baseline Score	0.5911 (BLEU) Japanese → English			
Baseline Score	0.1704 (BLEU) English → Japanese			

Table 5.2: Experimental setup Japanese ↔ English

5.4.2 Baseline Pruning

The probability and variety threshold pruning approaches from section 5.3.1 served as a baseline. 10 different probability thresholds (0, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1) and 14 variety thresholds (1, 2, 3, 4, 5, 6, 8, 10, 15, 20, 50, 100, 200, 500) were used. A probability threshold of 0 always means that there is no pruning, while in these cases, a variety threshold of 500 also does not prune any phrase pairs (which could be the case for other phrase tables).

It is usual practice to combine both of these threshold approaches rather than using just one, e.g. use a probability threshold of 0.001 and simultaneously apply a variety threshold of 10. This generally gives better performance than relying on just one kind of threshold pruning. For the baseline, all 140 combinations of these threshold values were used. These combinations give a variety of translation scores at different sizes.

Figures 5.6 and 5.7 show these scores for both translation directions. In both cases, the final translation scores for different threshold settings fluctuate considerably. For that reason, the baseline score at each possible size was defined as the best score that was reached with equal or fewer phrase pairs than the given size in any of the tested combinations.

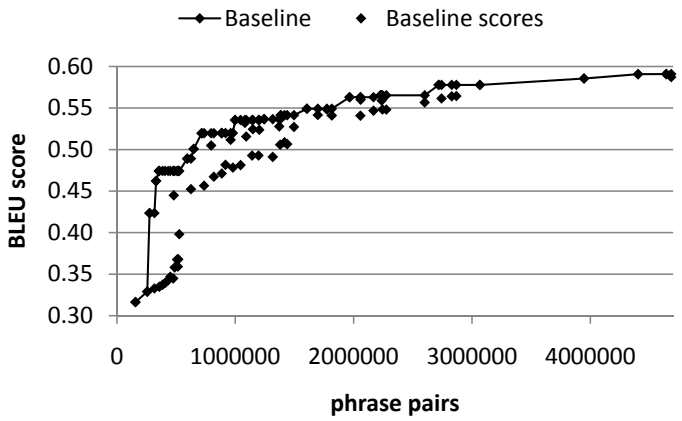


Figure 5.6: Baseline BLEU scores Japanese → English

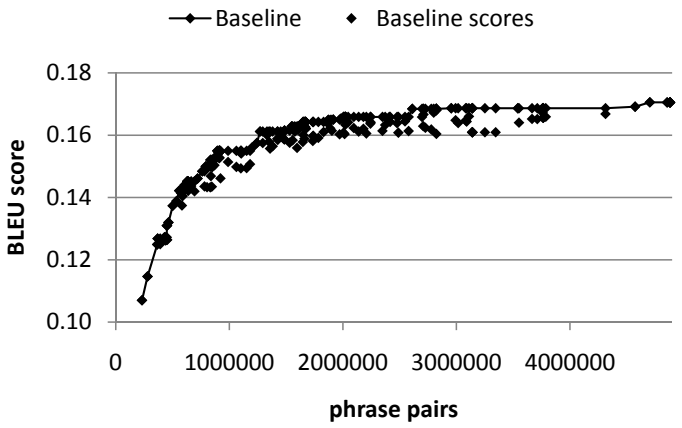


Figure 5.7: Baseline BLEU scores English → Japanese

Generally, when using those two thresholds it will be necessary to try various options to find the best combination for a given size. The scores fluctuate slightly less for the translation direction English → Japanese in figure 5.7, but here the baseline was also defined as the maximum score at each size. The table 5.3 shows a selection of BLEU scores with the resulting relative phrase table sizes. This table illustrates that the variety threshold generally gives better results than the probability threshold.
The presented pruning strategies were first investigated for the Japanese → English translation system to find the best pruning method and then applied to the English → Japanese system to validate the results and conduct further experiments.

	Probability threshold			
	0	0.0001	0.001	0.01
1	0.4901 (8.2%)	0.4743 (7.6%)	0.4621 (7.1%)	0.4235 (5.9%)
2	0.5196 (15.3%)	0.5008 (13.9%)	0.4891 (12.7%)	0.4449 (10.3%)
3	0.5355 (21.3%)	0.5198 (18.9%)	0.5049 (17.0%)	0.4524 (13.4%)
4	0.5523 (26.6%)	0.5320 (23.1%)	0.5117 (20.5%)	0.4565 (15.7%)
5	0.5543 (31.1%)	0.5366 (26.6%)	0.5157 (23.3%)	0.4673 (17.5%)
6	0.5585 (35.2%)	0.5414 (29.6%)	0.5238 (25.7%)	0.4712 (18.9%)
8	0.5630 (42.0%)	0.5491 (34.3%)	0.5278 (29.3%)	0.4782 (20.9%)
10	0.5653 (47.6%)	0.5482 (37.9%)	0.5272 (32.0%)	0.4812 (22.3%)
15	0.5779 (58.0%)	0.5601 (44.0%)	0.5416 (36.3%)	0.4928 (24.4%)
20	0.5777 (65.5%)	0.5589 (47.8%)	0.5410 (38.8%)	0.4928 (25.6%)
50	0.5856 (84.3%)	0.5567 (55.5%)	0.5408 (43.9%)	0.4911 (28.1%)
100	0.5907 (94.0%)	0.5615 (58.6%)	0.5468 (46.3%)	0.5058 (29.5%)
200	0.5909 (99.1%)	0.5641 (60.4%)	0.5482 (47.9%)	0.5088 (30.2%)
500	0.5911 (100.0%)	0.5644 (61.2%)	0.5481 (48.6%)	0.5065 (30.7%)

Table 5.3: BLEU scores after pruning at variety/probability thresholds with relative phrase table size in parentheses (variety thresholds in rows).

5.4.3 Recombination Pruning

The re-combination pruning approach was only tried for selected combinations of the probability and variety thresholds as the computational complexity is very high. The r-ratio threshold for the re-combination pruning was set to values of 1.1, 2, 5, 10 and 1000. An r-ratio of 1.1 means that the combined probability of the shorter phrase pairs has to be very close to the probability of the longer phrase pair. (An r ratio of 1 would require them to be equal, but that is unlikely due to rounding errors and the approximation of the phrase pair probabilities). An r-ratio threshold of 1000 means that every possible phrase re-combination will be used, disregarding the probabilities for the given phrase pairs. Table 5.4 shows the results for a probability threshold of 0 (no pruning) and variety thresholds of 5 and 10 while table 5.5 shows the same results with a probability threshold of 0.001. The tables list the percentage of phrase pairs that remain after pruning in parentheses. In both cases only a relatively small amount of phrases (about 1-2%) can actually be removed using this approach. It is interesting to note that the BLEU scores do basically not change. Even for the very large r-ratio threshold of 1000, the BLEU scores do not decrease significantly. This means that longer phrase pairs can almost always be replaced by shorter phrase pairs

r-ratio	variety threshold 5	variety threshold 10
none	0.5543 (31.1%)	0.5653 (47.6%)
1.1	0.5497 (31.0%)	0.5671 (47.5%)
2	0.5510 (30.7%)	0.5669 (47.0%)
5	0.5506 (30.5%)	0.5662 (46.7%)
10	0.5504 (30.4%)	0.5647 (46.6%)
1000	0.5495 (30.3%)	0.5598 (46.3%)

Table 5.4: Re-combination pruning with no probability pruning (threshold 0) and variety thresholds 5 and 10 (relative phrase table size in parentheses)

r-ratio	variety threshold 5	variety threshold 10
none	0.5157 (23.3%)	0.5272 (32.0%)
1.1	0.5152 (23.2%)	0.5255 (31.9%)
2	0.5164 (23.0%)	0.5256 (31.5%)
5	0.5150 (22.8%)	0.5289 (31.2%)
10	0.5136 (22.7%)	0.5289 (31.1%)
1000	0.5132 (22.6%)	0.5245 (31.0%)

Table 5.5: Re-combination pruning with probability pruning (threshold 0.001) and variety thresholds 5 and 10 (relative phrase table size in parentheses)

without significant performance loss, even if the probabilities do not match. On the other hand, the results are disappointing, as only a small number of phrase pairs can actually be eliminated using this technique.

5.4.4 Pruning via Usage Statistics

The statistics-based pruning strategies first collect statistics for actual phrase usage by translating large amounts of data. In this case the translation system was used to translate the 162,318 lines of Japanese training data. For each sentence, a 1000-best list was generated and the statistics were collected.

Empirical Score The empirical score ranks phrase pairs according to this equation:

$$score_{empirical}(\text{phrase pair}) = [\log(c(\text{phrase pair} + 1))] * [u(\text{phrase pair}) + 1]$$

Figure 5.8 and table 5.6 show the results for various sizes compared to the defined baseline.

♯phrase pairs	Baseline	Empirical Score	Relative Improvement
100,000	-	0.4735	-
200,000	0.3162	0.5008	58.38%
300,000	0.4235	0.5154	21.70%
400,000	0.4743	0.5241	10.50%
500,000	0.4743	0.5269	11.09%
600,000	0.4890	0.5359	9.59%
800,000	0.5194	0.5394	3.85%
1,000,000	0.5355	0.5442	1.62%
1,200,000	0.5366	0.5461	1.78%
1,500,000	0.5413	0.5523	2.03%
2,000,000	0.5630	0.5749	2.11%
3,000,000	0.5778	0.5798	0.35%
4,000,000	0.5855	0.5865	0.17%
4,684,044	0.5911	0.5911	0.00%

Table 5.6: Results for empirical scoring term and relative improvement vs. baseline

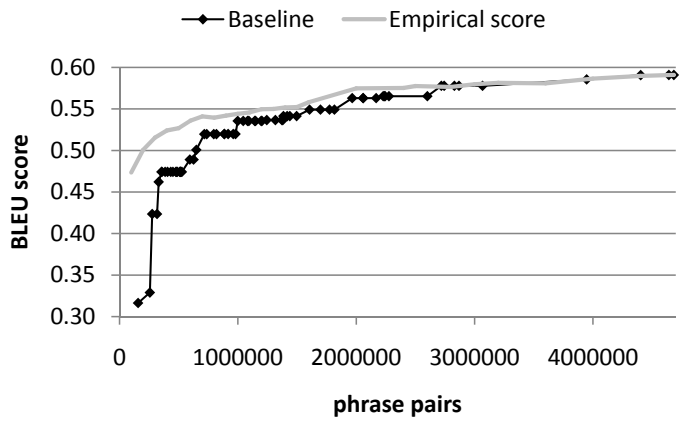


Figure 5.8: Empirical score vs. Baseline - BLEU scores Japanese → English

Very significant score improvements are realized for small phrase table sizes. For larger phrase tables, the improvements are naturally lower. The highest relative improvement over the baseline is 58% at 200,000 phrase pairs (None of the baseline pruning approaches, probability or variety threshold, reached 100,000 phrase pairs). At 2 million phrase pairs, the scores are

within the confidence interval of the full phrase table, but the improvements for phrase table sizes over 1 million are small and generally stay below 2%.

Model-best Path Pruning The model-best path pruning counts the number of occurrences of each phrase pair in the 10-best lists and assigns scores to each phrase pair according to the scoring terms $score_{A1}$ and $score_{A2}$ presented in section 5.3.2.6.

$$score_{A1} = \sum_{n=1}^{10} \frac{\sharp(\text{phrase pair in i-best})}{i}$$

$$score_{A2} = \sum_{n=1}^{10} \frac{\sharp(\text{phrase pair in i-best})}{i^2}$$

The results in table 5.7 show that both scoring terms clearly outperform the baseline pruning and the empirical score.

♯phrase pairs	Baseline	Empirical Score	$score_{A1}$	$score_{A2}$
100,000	-	0.4735	0.4792	0.4909
200,000	0.3162	0.5008	0.5306	0.5388
400,000	0.4743	0.5241	0.5596	0.5576
800,000	0.5194	0.5394	0.5747	0.5748
1,200,000	0.5366	0.5498	0.5788	0.5790

Table 5.7: Results for scoring terms $score_{A1}$ and $score_{A2}$

The scoring term $score_{A2}$ has a very small advantage over the scoring term $score_{A1}$. In both cases the score at 800,000 phrase pairs is already within the 95% confidence interval of the final BLEU score. The biggest difference can be observed at 100,000 phrase pairs. Here the score improves from 0.4792 to 0.4909. For larger sizes the scores are quite similar.

Figure 5.9 illustrates the improvements, comparing the baseline scores with the empirical score and $score_{A2}$.

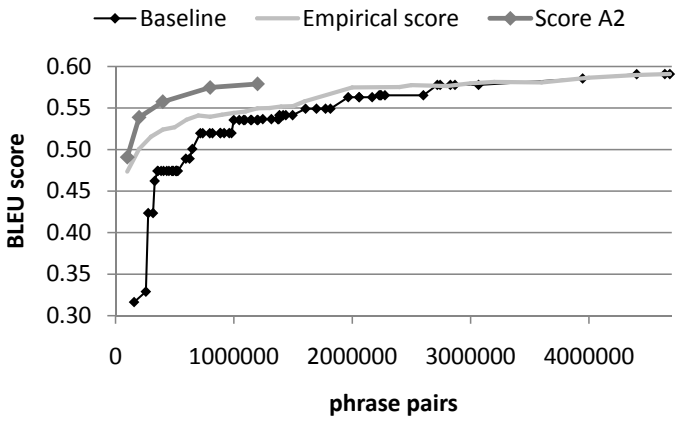


Figure 5.9: Improvements with $score_{A2}$ on Japanese → English

Metric-best Path Pruning The metric-best path pruning applies the same ideas but uses the number of occurrences in the top 10 metric-best paths to calculate the phrase pair scores (see section 5.3.2.7). In this experiment the metric used was edit distance compared to the references (minimum edit distance for multiple references).

$$score_{B1}(\text{phrase pair}) = \sum_{n=1}^{10} \frac{\sharp\text{phrase pair in metric-i-best}}{i}$$

$$score_{B2}(\text{phrase pair}) = \sum_{n=1}^{10} \frac{\sharp\text{phrase pair in metric-i-best}}{i^2}$$

Comparing these results in table 5.8 with the scores of the model-best path pruning in table 5.7 shows only minor differences. In this case $score_{B1}$ performs slightly better than $score_{B2}$.

Score Combination Because both pruning approaches performed very well separately, the scores were combined to take advantage of the benefits of both methods. The new scores are defined as the sum of these scores. Table 5.9 shows the results. Unfortunately, none of the combinations show a significant improvement over using a single method.

The main reason for this behavior is that for many sentences in the training data, the model-best translation is also the metric-best translation (or they are very close). This would be unusual for unseen test data, but in this case, the phrases were originally extracted from this data. Therefore, the

♯phrase pairs	Baseline	Empirical Score	$score_{B1}$	$score_{B2}$
100,000	-	0.4735	0.4822	0.4759
200,000	0.3162	0.5008	0.5300	0.5266
400,000	0.4743	0.5241	0.5610	0.5572
800,000	0.5194	0.5394	0.5753	0.5654
1,200,000	0.5366	0.5498	0.5787	0.5707

Table 5.8: Results for scoring terms $score_{B1}$ and $score_{B2}$

♯phrase pairs	$score_{A1} + score_{B1}$	$score_{A1} + score_{B2}$	$score_{A2} + score_{B1}$	$score_{A2} + score_{B2}$
100,000	0.4786	0.4852	0.4883	0.4909
200,000	0.5306	0.5340	0.5395	0.5386
400,000	0.5596	0.5597	0.5585	0.5562
800,000	0.5747	0.5752	0.5747	0.5751
1,200,000	0.5791	0.5791	0.5791	0.5790

Table 5.9: Results for combination of scoring terms

model-best statistics will only differ slightly from the metric-best statistics. Consequently, this resulted in those very similar scores in table 5.7 and table 5.8. The combination score will then not considerably change the order of phrase pairs, which was also confirmed in a separate analysis.

Altogether, scoring term $score_{A2}$ seems to be the best choice for practical applications. It gives consistently good results without using the metric-best path information.

Pruning towards the Metric-best Path To test the last approach proposed in section 5.3.2.8, the additional statistics $score_E$ was used. As described, this statistic aims to find phrase pairs that occur in a path that gets a higher model score than the metric-best path, thereby preventing the metric-best path from being chosen:

$score_E$(phrase pair) = Number of times the phrase pair occurs in a path that has a higher model score than the metric-best path while *not* occurring in the metric-best path.

To eliminate these phrase pairs, this score was subtracted from the $score_{A2}$. Just subtracting it gave low scores, so its influence was limited

by adding factors of 0.1, 0.01 and 0.001. The results in table 5.10 show that a factor of 0.01 seems to have slight advantages for lower numbers of phrase pairs, but it can generally be stated that this additional score does not significantly improve the overall performance.

♯phrase pairs	$score_{A2} - 0.1 \cdot score_E$	$score_{A2} - 0.01 \cdot score_E$	$score_{A2} - 0.001 \cdot score_E$	$score_{A2}$
100,000	0.4841	0.4911	0.4909	0.4909
200,000	0.5371	0.5392	0.5386	0.5388
400,000	0.5536	0.5565	0.5576	0.5576
800,000	0.5711	0.5740	0.5747	0.5748
1,200,000	0.5777	0.5792	0.5792	0.5790

Table 5.10: Results when incorporating $score_E$

Amount of Data used to Estimate Statistics It can be relatively tedious to translate all 162,318 lines of data to collect the statistics for the pruning, so smaller amounts of data were also used to estimate the statistics. This way it is possible to determine, how the pruning may be affected by a smaller data set.

Table 5.11 compares the results when using 40,000 and 80,000 lines of data (randomly chosen) with the translation of the full training data of 162,318 lines. In both cases, the results drop very significantly and the additional data in the full training corpus helps tremendously.

♯phrase pairs	$score_{A2}$ on 40k	$score_{A2}$ on 80k	$score_{A2}$ on 162k
100,000	0.4032	0.4159	0.4909
200,000	0.4377	0.4611	0.5388
400,000	0.4488	0.4896	0.5576
800,000	0.4661	0.5072	0.5748
1,200,000	0.4854	0.5204	0.5790

Table 5.11: Results with different data sizes to estimate the statistics

Re-estimating Statistics during Pruning All previous experiments followed a strict sequence. In the first step, the translation model is used to translate the large amount of data. During the translation, the statistics are

collected that allow the scoring, sorting and pruning of phrase pairs. An alternative approach is pruning in batches. At first, only a certain number of phrase pairs are actually removed. The partly pruned phrase table is then again used to translate the data and estimate new statistics to remove the next batch of phrase pairs. The advantage of this approach could be that the translation system might significantly change its phrase usage with the already pruned phrase table. This could make it beneficial to estimate new statistics. The disadvantage is the increased complexity, as it is necessary to re-translate a large amount of data multiple times. This experiment was done using batches of 100,000 and 50,000 phrase pairs that would be pruned at one time. The results in table 5.12 reveal that all improvements gained here are not significant. There is hardly any change for the larger phrase tables and only a slight improvement for the smaller sizes.

♯phrase pairs	$score_{A2}$	$score_{A2}$ 50k batch	$score_{A2}$ 100k batch
100,000	0.4909	0.4924	0.4917
200,000	0.5388	0.5397	0.5394
400,000	0.5576	0.5578	0.5582
800,000	0.5748	0.5747	0.5750
1,200,000	0.5790	0.5791	0.5788

Table 5.12: Results after repeated re-estimation of the statistics

5.4.5 Experiments on English → Japanese

For validation purposes and additional experiments, the pruning strategies were also applied to translations from English to Japanese. To collect the statistics the English part of the bilingual training data of 162,318 lines, was translated to Japanese using the originally extracted phrase pairs. The results are very similar to the other experiments. $score_{A2}$ also outperforms the empirical score and the baseline pruning (table 5.13 and figure 5.9).

♯phrase pairs	Baseline	Empirical Score	$score_{A2}$
100,000	-	0.1366	0.1421
200,000	0.1011	0.1444	0.1540
400,000	0.1250	0.1512	0.1581
800,000	0.1500	0.1562	0.1652
1,200,000	0.1559	0.1616	0.1672

Table 5.13: Results for English → Japanese

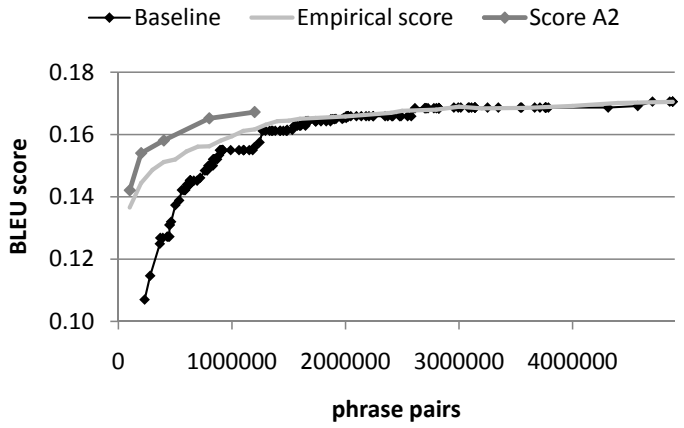


Figure 5.10: Improvements with $score_{A2}$ on English → Japanese

Influence of Additional Data The last experiment tested how additional out-of-domain data affected the performance of the pruning. For this purpose, the translation system was used to translate 40,000 lines of English medical dialog data in addition to the bilingual data to estimate the statistics. The style of the medical dialog data is not very different from the

BTEC data, but the topics are obviously out-of-domain. The results in table 5.14 show that this additional data did not help the pruning performance significantly, and some of the numbers are actually slightly lower.

♯phrase pairs	$score_{A2}$	$score_{A2}$ + 40k medical data
100,000	0.1421	0.1415
200,000	0.1540	0.1535
400,000	0.1581	0.1573
800,000	0.1652	0.1655
1,200,000	0.1672	0.1671

Table 5.14: Using additional data to estimate statistics

It could be necessary to have a significant amount of additional in-domain data available to further improve these statistics, but this was not available at the time.

5.4.6 Further Analysis

The previous results showed very significant improvements over the strong baseline pruning approaches. This section will further analyze how this improvement was possible and point out the main reasons.

Multiple Translation Candidates The fundamental question is how a source phrase can end up with a large number of translation candidates. Generally, there are three possible situations that can have this effect:

- Frequent occurrence
- Long phrase pair
- Ambiguous phrase pair

A phrase that occurs frequently will co-occur with a larger number of other words and, therefore, has the potential to be aligned with a larger number of words. Typical examples for this are function words like “and” and “the”.

Another possibility is a long phrase pair where various alignment possibilities exist. These can add up to more translation candidates.

The last option is an actually ambiguous phrase. Ambiguous phrases might be translated differently in the sentences in the bilingual training data

and this leads to a large number of translation candidates. An example here could be the word "put", which has many different meanings that could correspond to different words in other languages (compare section 4.3.2).

Overall, all three situations will lead to multiple translation candidates, but only the last case actually justifies and requires them. In the first two cases, the large number of translation candidates is merely an artifact of the alignment models.

Standard and Proposed Approaches The problem with the standard pruning approaches is that all these cases are treated exactly the same. If a variety threshold of 5 is imposed, it will cut the number of translations regardless.

The proposed approaches based on usage statistics are more adaptive to these situations. For example, the word "Tokyo" will occur frequently in the English/Japanese data and likely get a large number of translation candidates. However only one (or a few) translations are frequently used during the collection of the statistics, so the other candidates are pruned.

In case of an ambiguous word like "put" more translation candidates will be used so fewer phrase pairs will actually be pruned.

Comparing Candidate Distributions The situation also becomes clear when considering the distribution of the number of target phrase pairs per source phrase. Figure 5.11 shows the situation before pruning the phrase table. Each bar represents the number of source phrases that have this many

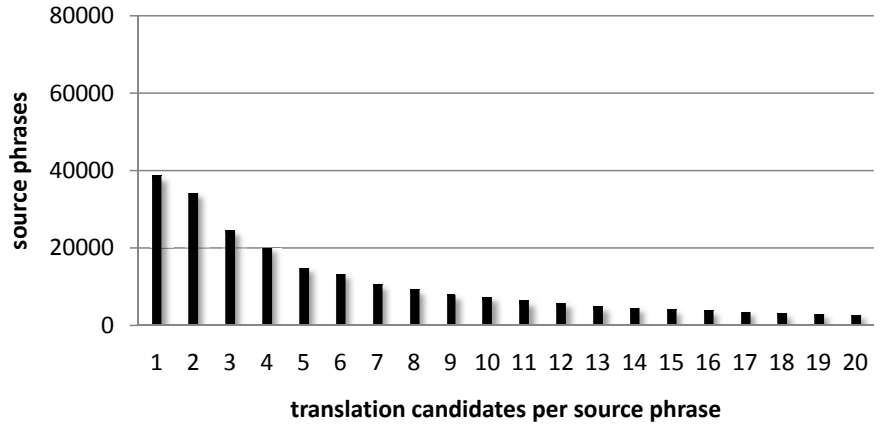


Figure 5.11: Translation candidates for source phrase pairs - baseline distribution

translation candidates. Most of the source phrases have only one candidate, but low numbers are also common. A steady drop can be observed until 20 translation candidates, as shown in the figure.

If a variety threshold of 10 is used, the distribution becomes as displayed in figure 5.12.

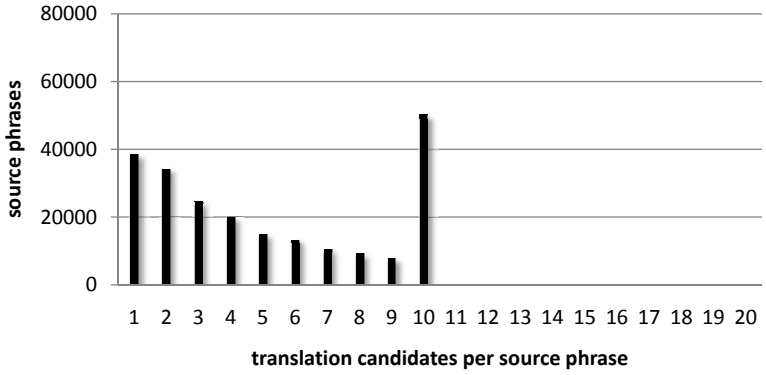


Figure 5.12: Translation candidates for source phrase pairs - variety threshold 10

There is no change in the bars for one to nine translation candidates, but all source phrases that had eleven or more translation candidates are now added to the bar for ten candidates. Consequently, a very large number of source phrases now has ten candidates and no source phrases have more candidates. It is already obvious that this creates a very unbalanced situation.

Using the proposed approach ($score_{A2}$) the situation is very different. Figures 5.13, 5.14, 5.15 and 5.16 show the distributions after sorting and limiting the overall number to 1.2 million, 400,000, 200,000 and 100,000 phrase pairs. It is clear that the number of source phrases with one translation candidate increases rapidly, but there are always source phrases left with larger numbers of translation candidates. Even at 100,000 phrase pairs, there is still a significant number of source phrases with more than one translation candidate left.

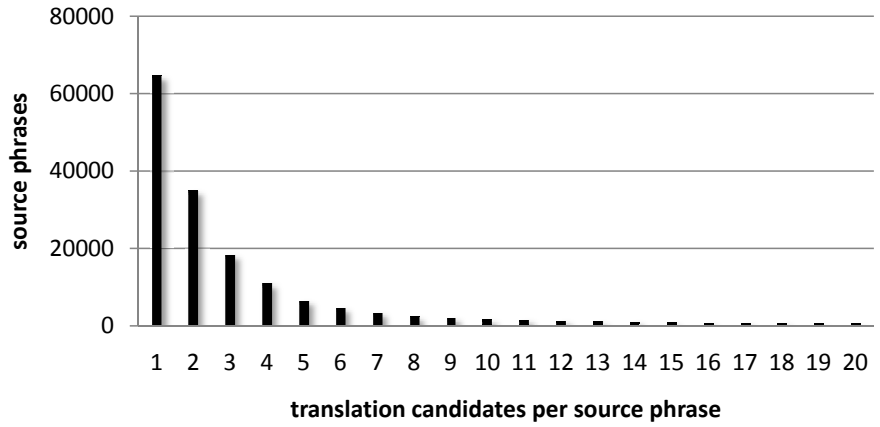


Figure 5.13: Translation candidates for source phrase pairs - proposed pruning 1.2 million phrase pairs

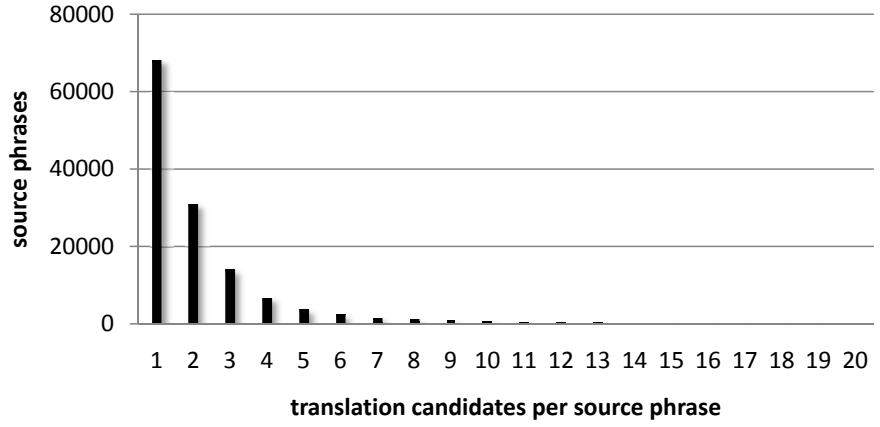


Figure 5.14: Translation candidates for source phrase pairs - proposed pruning 400,000 phrase pairs

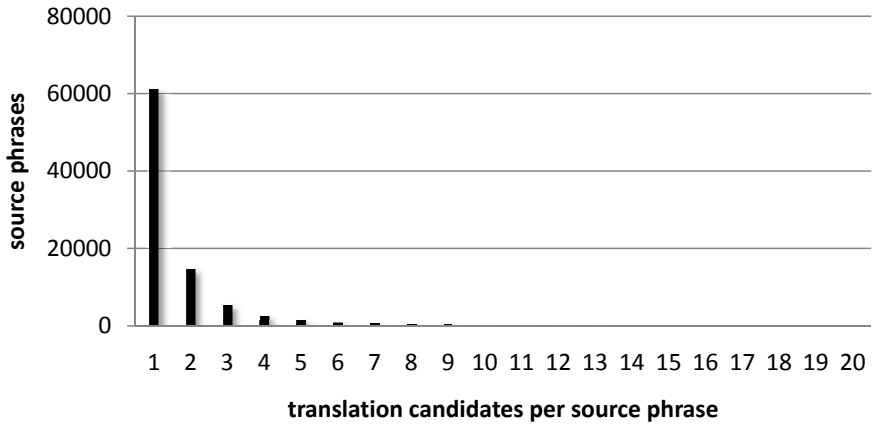


Figure 5.15: Translation candidates for source phrase pairs - proposed pruning 200,000 phrase pairs

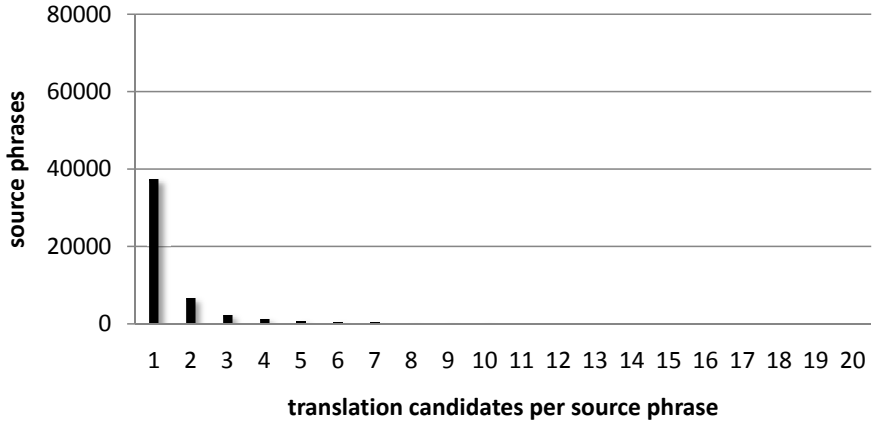


Figure 5.16: Translation candidates for source phrase pairs - proposed pruning 100,000 phrase pairs

Additional Phrase Table Compression One potential problem a pruning based approach could incur is a minimized possibility to further compress the phrase table. All compression techniques are based on data redundancy and redundancy is also exploited in the presented approaches. To test how much additional compression would still be possible, the baseline and pruned phrase tables in text format were compressed using the gzip (`www.gnu.org/software/gzip`) tool which implements the DEFLATE) algorithm, a combination of LZ77 (Ziv and Lempel, 1977) and Huffman coding (Huffman, 1952). The results in table 5.15 do indeed show a smaller size reduction for the pruned phrase tables, but the differences are minor. Even for only 100,000 phrase pairs, the possible reduction is only 8% smaller than for the full phrase table with 4.6 million phrase pairs.

phrase pairs	gzip reduction
4.6M	-63.7%
1,200,000	-58.9%
400,000	-56.3%
200,000	-55.8%
100,000	-55.7%

Table 5.15: gzip reduction of pruned phrase tables

5.5 Conclusions

The proposed pruning approaches show very significant improvements over a strong baseline that was defined as the maximum of the standard probability and variety threshold methods. The empirically found score is already able to outperform the baseline, but incorporating additional analysis could further improve the results. Overall, it is possible to remove up to 80% of phrase pairs while not significantly affecting the translation performance. This allows the translation models to be put on much smaller and mobile devices.

The computational effort is quite high, as large amounts of data have to be translated to collect the necessary statistics. However, the threshold pruning also requires a large computational effort, as multiple combinations of thresholds must be evaluated to find the best fit.

Using the metric-best path information did not provide additional benefits compared to the model-best paths. It was also not notably valuable to try to enforce the metric-best path by eliminating certain phrase pairs that might prevent this path from being chosen. There are two likely reasons for this

behavior. The translations were done on the training data, so the metric-best path often does not differ from the model-best path. It could be interesting to design a method that splits the training data in various parts and estimates the statistics on held-out parts.

However, for many sentences the model-best path and metric-best paths do differ. Just keeping the phrase pairs from the metric-best path does not actually enforce them, so no significant improvement can be observed, but it was also not possible to eliminate supposedly bad phrase pairs just based on the metric-best path. This also means, that it might not be possible with this method to classify phrase pairs into two classes, "good" ones and "bad" ones.

It is definitely advisable to use the full bilingual corpus to extract the statistics. The scores dropped significantly if only part of the data was used. Using additional data from another domain did not improve the scores.

The repeated re-translation of the data for the pruning in batches approach did not prove valuable. Apparently the system does not need to adjust to the changed situation, as the top phrase pairs are not pruned.

Using the proposed approaches, it is possible to completely remove source phrases. This could negatively affect the overall performance in practical applications, especially if source vocabulary is dropped. It would, however, be trivial to keep at least one translation candidate for each word or source phrase, even if the proposed approach would eliminate it.

Chapter 6

Improving Vocabulary Coverage

6.1 Introduction

It was already pointed out in section 3.4 that the vocabulary coverage of current translation systems is not fully sufficient for actual usage. This mainly concerns named entities, but other words are also affected, particularly technical and specialty terms.

Typical conversations in all three domains discussed here will almost always involve some kind of named entity. A tourist will ask for a flight or directions to a specific destination. He will inquire about restaurants, hotels, street names etc. and also get answers containing named entities from these categories. In medical applications, different medicines, symptoms and diseases are frequent in every conversation. The same situation arises for military users. Table 6.1 lists common categories for named entities and specialty terms from these domains, but there are certainly more. Also, every

Tourism	Medical	Military
Cities	Medicine	Cities
Neighborhoods	Symptoms	Neighborhoods
Streets	Diseases	Streets
Sights	Chemicals	Bases
Restaurants	Body Parts	Civil Engineering
...	...	...

Table 6.1: Named entity/specialty term categories in different domains

other domain will most likely have a similar need for named entities and specialty terms. The problem becomes even more severe as each individual person using a translation system has individual needs. Every tourist might have different interests, hobbies or preferences that influence his vocabulary. A model railroad hobbyist will also be interested in model railroads while he is on vacation in a foreign country. Collectors might want to add foreign items to their collections. These countless hobbies and interests have, again, thousands of specific technical terms, and the ability to use them will be important to the individual user.

Medical and Military users have job specializations with various technical terms and these could actually be crucial for their individual mission or function. The medical profession in particular has thousands of specialized terms and often multiple terms for one concept, a technical term and a common name (e.g. "sternum" and "breastbone" refer to the same concept).

This thesis will at first not specifically distinguish between named entities and specialty terms, but try to overcome the limited coverage with similar approaches.

Overview The next section 6.2 will demonstrate how the general name and specialty term coverage of a translation system can be improved. This will often require manual work, as large name lists have to be collected. In this case, a medical database was available that offered large medical name lists, so no manual labor was necessary.

The large name lists pose an additional problem if they are supposed to be used on a mobile device with limited memory. A method will be proposed in section 6.3 that will improve the handling of large name lists in mobile devices. The goal is to produce personalized translation models for each individual user, based on specific needs and interests.

Even with increased name and specialty term coverage it will never be possible to cover *all* named entities and specialty terms and the translation systems will always be confronted with unknown words. Section 6.4 will present an new approach to "communicate" unknown words. In summary, three tasks will be discussed here:

1. Improving name and specialty term coverage
2. Handling large name lists on mobile devices
3. Dealing with the remaining unknown words

Relevant related work for the first two tasks is rather limited. However, the translation of unknown words is a common problem. Here extensive research has been done, and various publications are available that will be discussed.

6.2 Improving Vocabulary Coverage

The first goal is to improve the general named entity and specialty vocabulary coverage. Simply collecting more bilingual training data will not solve this problem. It will neither be efficient nor possible to collect data that will really cover *all* relevant named entities and technical terms just by adding more and more documents.

It will be far more valuable to separately collect and produce bilingual name lists and use these name lists in the overall translation system. This will allow the separation of the actual phrase translation knowledge in the bilingual data from the purely dictionary based named entity translations in the collected name lists. A city name list will allow to replace "Tokyo" in a sentence like "I would like to fly to Tokyo". Assuming the bilingual data contains the correct phrase pairs any city name will produce a generally correct translation.

These name lists cannot be collected completely automatically from bilingual or monolingual data. Named entity taggers are available and can also classify the tagged named entities, but their performance is still limited (Downey et al., 2007; Collins and Singer, 1999). At the very least a manual checking of the name lists would be necessary if documents or webpages are the name source.

If specialized databases are available, it is occasionally possible to generate large name lists completely automatically. GIS (geographic information system) databases contain information about locations, cities, street names and sights. Phone books might also contain street names and person names. Most of these databases will only contain the names in one language so the name list will have to be translated.

This thesis will not discuss the manual labor necessary to acquire these name lists in detail. A name list will also have to be translated in full. After eliminating duplicate names, repetitions will be unlikely. A simple frequency based sorting might, however, be valuable.

What will be discussed is how available name lists can be effectively applied. In the experiments in the next section, a medical database was used that did allow the extraction of bilingual name lists directly for various medical classes, so no manual labor was necessary to produce and translate the name lists.

6.2.1 Medical Terminology

The first focus will be on improving the vocabulary coverage in the medical domain. The same applies to the military and tourism domains, but for these

no large bilingual name lists were readily available.

For Medical named entities, beneficial electronic document sources would be popular medical websites like WebMD (`www.webmd.com`), MayoClinic (`www.mayoclinic.com`), MedicineNet (`www.medicinenet.com`) or databases for medical publications like MEDLINE available via PubMed (`www.pubmed.org`).

A freely available medical database is the Unified Medical Language System (UMLS, `www.nlm.nih.gov/research/umls`) that was used for some experiments to extend the named entity and specialty vocabulary coverage of a Spanish ↔ English translation system. The results of these experiments were published in Eck et al. (2004). Those experiments were originally done using the 2003AB version of the UMLS and were re-done with the more recent version 2007AC. The UMLS database is continuously being expanded and is part of an ongoing research effort to add more vocabularies and languages.

6.2.2 Experiments with the Unified Medical Language System

6.2.2.1 Unified Medical Language System

The UMLS project was initiated in 1986 by the U.S. National Library of Medicine. It integrates different knowledge sources like biomedical vocabularies and dictionaries into one database. The goal is to help health professionals and researchers by combining biomedical information from these different sources and making them easily accessible. It is usually updated about three or four times per year. The UMLS consists of three main knowledge repositories, the UMLS Metathesaurus, the UMLS Semantic Network and the SPECIALIST lexicon. Additional facts about the UMLS, related work and further information can be found in Browne et al. (2003), Friedman et al. (2001), Kashyap (2003), Lindberg (1990) and at `www.nlm.nih.gov/research/umls`.

UMLS Metathesaurus The UMLS Metathesaurus provides a common structure for approximately 100 source biomedical vocabularies. The 2007AC version of the Metathesaurus contains 1,516,299 concepts named by 7,426,224 terms (case sensitive). It is organized by concept, which is a cluster of terms (i.e. synonyms, lexical variants and translations) with the same meaning. Translations are present for up to 16 additional languages besides English. It is very likely that other languages will be added in later releases.

Table 6.2 shows the distribution of the terms according to the 17 different languages in UMLS 2007AC. Most terms are English, but a number of other languages also have a considerable vocabulary.

Language	Number of Terms	Language	Number of Terms
English	4,729,931	Swedish	32,542
Spanish	1,529,152	Finnish	25,079
Dutch	215,644	Danish	723
German	168,938	Norwegian	722
French	157,228	Hungarian	718
Portuguese	144,181	Basque	695
Italian	112,241	Hebrew	485
Russian	50,553		

Table 6.2: Languages in the UMLS

For example the concept “arm” includes the English lexical variant, its plural form, “arms” and with “bras”, “arm”, “braccio”, “braço”, “ruka” and “brazo” the French, German, Italian, Portuguese, Russian and Spanish translations (among others).

Entries are not limited to single words and some terms are also longer phrases like “third degree burn of lower leg” or “loss of consciousness”. It also includes inter-concept relationships across the multiple vocabularies. The main relationship types are listed in Table 6.3:

Relationship types
broader
narrower
like
parent
child
sibling
is allowed qualifier
can be qualified by
is co-occurring with
other related

Table 6.3: Relationships in the UMLS

The synonym-relationship is implicitly realized by different terms that are affiliated with the same concept. The co-occurrence relationship refers to concepts co-occurring in MEDLINE publications.

The UMLS Semantic Network The UMLS Semantic Network categorizes the concepts of the UMLS Metathesaurus through semantic types and relationships. Every concept in the Metathesaurus is part of one or more semantic types. There are 135 semantic types arranged in a generalization hierarchy with the two roots "Entity" and "Event". This hierarchy is still rather abstract and not deeper than six (see table 6.4). A more detailed generalization hierarchy is realized with the child, parent and sibling relationships of the UMLS Metathesaurus.

Entity
- Physical Object
  - Organism
  - Anatomical Structure
    - Fully Formed Anatomical Structure
      - Body Part, Organ or Organ Component
  - Manufactured Object
    - Medical Device
      - Drug Delivery Device
  - Clinical Drug

Event
- Activity
  - Behavior
    - Social Behavior
  - Occupational Activity
    - Health Care Activity
      - Laboratory Procedure
- Phenomenon or Process
  - Human caused Phenomenon or Process

Table 6.4: Semantic types in the UMLS - Examples

The SPECIALIST Lexicon The SPECIALIST lexicon contains over 330,000 English words. It is intended to be a general English lexicon including many biomedical terms. The lexicon entry for each word or term records the syntactic, morphological and orthographic information. Table 6.5 shows the entry for "anesthetic". There is a spelling variant "anaesthetic" and an entry number. The category in this case is noun (there is another entry for "anesthetic" as an adjective). The variants slot contains a code indicating

the inflectional morphology of the entry. “anesthetic” can either be a regular count noun (with regular plural “anesthetics”) or an uncountable noun.

```
{base=anesthetic
spelling_variant=anaesthetic
entry=E0330018
        cat=noun
        variants=reg
        variants=uncount
}
```

Table 6.5: Specialist Lexicon in the UMLS - Example “anesthetic”

6.2.2.2 Extracting Dictionaries from the UMLS

The first way to exploit the UMLS database in order to increase the name coverage is naturally to extract additional Spanish ↔ English lexicons and phrase books for the named entities in the UMLS. The UMLS Metathesaurus provides translation information, as it can be assumed that Spanish and English terms that are associated with the same concept are respective translations. For example, as the English term “arm” is associated with the same concept as the Spanish term “brazo” it can be deduced that “arm” is the English translation of “brazo”.

Unfortunately the UMLS does not contain morphological information about languages other than English at the moment. This means it cannot be automatically detected that “brazo” is the singular form and thus the translation of “arm” and not the translation of “arms”. As most of the entries are in singular form, every possible combination of Spanish and English terms was extracted regardless of possible errors like combining the singular “brazo” and the plural “arms”.

The resulting Spanish ↔ English lexicon/phrasebook for named entities contains 5,001,195 pairs of words and phrases (after conversion to lower case and some post processing). This is a very significant addition to any Spanish ↔ English translation system and should improve the named entity performance. However, a large amount of these translation pairs is made up of synonyms that drastically increase the overall number and some of the terms in the UMLS and the resulting lexicon might actually be too specific for a spoken language translation system like “1,1,1-trichloropropene-2,3-oxide” translating to “oxido de tricloropropeno”.

6.2.2.3 Generalizing the Training Data using UMLS Dictionaries

A more advanced way to use the UMLS database and name lists in other domains is to generalize the training sentences based on the extracted named entity list and the name categories. In this case the training data contains sentence pairs like:

Necesito examinar su cabeza.	I need to examine your head.
Necesito examinar su brazo.	I need to examine your arm.
Necesito examinar su rodilla.	I need to examine your knee.

After generalizing these sentences by replacing the specific body parts like "head", "arm" and "knee" with a general tag e.g. "@BODYPART" all those example sentences can be joined into one sentence.

Necesito examinar su `@BODYPART.`	I need to examine your `@BODYPART.`

Only a name list is necessary to translate the individual body parts in order to be able to correctly translate all sentences of this type. Possibly unseen sentences like "Necesito examinar su antebrazo" ("I need to examine your forearm") could also be correctly translated, if it could be automatically deduced that "antebrazo/forearm" is a body part and this translation pair was known.
Some other similar sentences in which the same ideas could be applied:

El `@BODYPART` esta inflamado.	The `@BODYPART` is inflamed.
¿Que `@BODYPART` le/la duele?	Which `@BODYPART` hurts?

The second sentence on the Spanish side illustrates a problem. Some languages change other parts of the sentence depending on the grammatical gender of the body part or the gender of the person. As this error is unlikely to affect the translation of the meaning, it was decided to ignore this problem.

As stated before, every concept in the UMLS Metathesaurus is categorized into one or more semantic types defined in the UMLS Semantic Network. The two semantic types "Body Part, Organ, or Organ Component" and "Body Location or Region" from the UMLS Semantic Network cover pretty closely what is usually affiliated with the colloquial meaning of

body part[1]. Using this information, the general Spanish ↔ English dictionary originally extracted from the UMLS was filtered, to contain only words and phrases from the two semantic types "Body Part, Organ, or Organ Component" and "Body Location or Region". This gave a dictionary of 281,690 translation entries for body parts. In the next step, every occurrence of a word or phrase pair from this new dictionary in the training data is replaced with a general body part tag ("@BODYPART"). It must be ensured here that the translation pairs occur in both languages. A standard training of the translation system with this changed training data then results in phrase pairs containing this tag. This is completely transparent for the training programs at this point. The tag "@BODYPART" can be treated as a regular word.

Then the "cascaded phrase tables" technique is applied that was proposed in Vogel and Ney (2000). During the translation process, the first phrase table that is applied in this case is the body part dictionary. It replaces the Spanish body part with its translation pair and the body part tag "@BODYPART". The following phrase tables can now apply their generalized rules containing the "@BODYPART" tag instead of the real body part.

An example will illustrate the translation method using the cascaded phrase tables. Source sentence to be translated:

> Necesito examinar su antebrazo.

First step: Apply body part dictionary phrase pair (antebrazo → forearm)

> Necesito examinar su `@BODYPART`{antebrazo → forearm}.

Apply generalized phrase pair: (e.g.: Necesito examinar su @BODYPART → I need to examine your @BODYPART)

> I need to examine your `@BODYPART`{antebrazo → forearm}.

Finally, resolve tags:

> I need to examine your forearm.

[1]The terminological difference is that the semantic type "Body Part, Organ, or Organ Component" is defined by a certain function. For example "liver" and "eye" are part of this semantic type, whereas the semantic type "Body Location or Region" is defined by the topographical location of the respective body part. Examples are "head" and "arm". The function in this case is not as clearly defined as the function of a "liver".

6.2.2.4 Translation Experiments

The Baseline system, which was used to test different approaches to improve the translation performance, is a statistical machine translation system. The task was to facilitate doctor-patient dialogs across languages from Spanish to English.

Translation System All experiments were again done with a state-of-the-art statistical machine translation system (Vogel, 2003; Eck et al., 2006). The system uses the phrase extraction method PESA described in Vogel (2005) and a 6-gram language model (Zhang and Vogel, 2006).

Test and Training Data The system was trained using 9,227 lines of training data (90,012 English words, 89,432 Spanish words). 3,227 lines of this data are medical dialog data. The 6,000 other lines of training data are BTEC data.

The test data consists of 500 lines with 6,886 words. The test data was also taken from medical dialogs between a doctor and a patient and contains a reasonable number of medical terms, but the language is not very complex. Table 6.6 shows some example test sentences (from the reference data). The Baseline system scores a 0.1912 BLEU (see table 6.7).

	...
Doctor:	The symptoms you are describing and given your recent change in *diet*, I believe you may be *anemic*.
Patient:	*Anemic*? Really? Is that serious?
Doctor:	*Anemia* can be very serious if left untreated. Being *anemic* means your body lacks a sufficient amount of *red blood cells* to carry *oxygen* through your body.
	...

Table 6.6: Medical dialog: Example test sentences

Adding the Dictionary In the first step the extracted lexicon/phrasebook was added as an additional phrase table without using the cascaded phrase tables. The experiment showed a nice increase in BLEU performance and scored at 0.2015 BLEU. This system especially has a higher coverage, as only 302 words (types) are not covered by the training data,

Bilingual Training Data			Monolingual Training Data	
	English	Spanish		English
Lines	9,227	9,227	Lines	9,227
Words	90,012	89,432	Words	90,012
Translation Models	PESA phrase table			
Language Model	Suffix Array 6-gram			
Test Data	500 lines, medical dialogs			
Baseline Score	0.1912 (BLEU)			

Table 6.7: Experimental setup Spanish → English

compared to 411 for the baseline system. In this case the language model was not changed and treated all new words as unknown words.

Adding the English part of the extracted dictionary to the language model further increased the score to 0.2034 BLEU.

Using the Semantic Type Information Using the class based approach to use name lists, the extracted "@BODYPARTS" name list was chosen along with two additional classes: "Finding, Sign or Symptom" and "Disease" (325,673 and 987,460 translation pairs respectively). As these also occur frequently in the test data. This further improved the score to 0.2197 BLEU. Adding other classes did not further improve the score, as none of these appeared in the test data. All available classes should certainly be used for all practical applications.

Analysis Table 6.8 gives an overview of the BLEU results of the described experiments, using the UMLS database at different stages. Each step increases the score, adding up to a significant improvement.

System	BLEU
Baseline	0.1912
Added Dictionary	0.2015
Added to LM	0.2034
Semantic Types	0.2197

Table 6.8: Results overview

It is not surprising to gain significant improvements by adding a large additional phrase table and more language modeling data. However, as was

discussed at the beginning, the BLEU score is not necessarily a measurement for improvements regarding names and does not fully reflect the actual situation in this case.

The same argument is made in Huang (2006) and the author uses the information gain to measure the improvments when translating named entities. But it is also obvious – a translation system that can translate so many more names and specialty terms compared to the baseline system will be far more useful in real applications than the rather small score increase indicates. On the other hand, the cascaded phrase table approach also generates improvements in BLEU score compared to just using the data as an additional phrase table. This implies advances in fluency. No additional vocabulary is covered, so the gains could only have resulted from improved phrase selection and word order. The reason is that the cascaded approach can match longer phrases after the generic tags have been introduced.

6.2.3 Improving Coverage - Tourism and Military

For tourism and military domains, no database was available that would contain such a large number of named entities and specialty vocabulary even remotely comparable to the Unified Medical Language System.

As this is not available, data from other sources has to be used, and name lists have to be created semi-automatically or even manually. The main problem is finding sources that will contain these names and specialized vocabulary.

For tourism in general, travel guides relevant to the country contain large amounts of named entities - places of interest, hotel, restaurant, shopping and food information. Mapping companies that supply GPS systems such as LeadDog (`www.goleaddog.com`) and map websites such as Google Maps (`maps.google.com`) also contain large name databases for specific categories, mainly related to cities, street names, general locations, places of interest, and also hotels and restaurants. Online recipe databases and restaurant menus might offer a good starting point for food related named entities. Table 6.9 shows a selection of the most relevant categories for names lists in the tourism domain.

Relevant military categories are relatively similar to the tourism categories, especially categories related to locations. Here the same sources could be exploited. Military related websites might provide additional military specific terms and named entities. Military and defense publications like Jane's (`www.janes.com`) will also contain specialty vocabulary.

- Location
  - Country
  - State
  - County
  - Town
  - Neighborhood
  - Street
  - Square
  - Landscape Location
    - Mountain range
    - Mountain
    - Valley
    - Beach
    - Winter Sports Area
  - Body of Water
    - Ocean
    - River
    - Lake
    - Bay/Gulf
- Sight
  - Museum
  - Monument
  - Building
  - National Park/Park
- Hotel related
  - Hotel chain
  - Hotel
- Shopping Related
  - Store
  - Currency
  - Product
- Transport related
  - Airline
  - Airport
  - Train
  - Train station
  - Bus/Bus company
  - Bus station
  - Taxi/Taxi company
  - Taxi stand
  - Car rental company
- Food related
  - Restaurant
  - Dish
  - Drink
  - Candy
  - Meat
  - Fish
  - Vegetable
  - Fruit
  - Other food
- Religious
- Other
- Non-nouns

Table 6.9: Selected categories relevant for the tourism domain

6.2.4 Maintaining Coverage across Languages

It is important to differentiate named entities based on how specific to a country or language they are. Most translation systems between two languages will mainly need the city names in the respective countries or group of countries where the languages are spoken. Other city names will be less important.

This is an actual problem. The BTEC corpus contains a small number of named entities, primarily from Japan, as it originated from Japanese phrase books. Named entities from other countries are very rare. However, this

means that if the BTEC corpus is translated to Spanish and a Spanish ↔ English translation system is trained on this data, it will not be as valuable as the Japanese ↔ English system. It will contain the same names as the Japanese ↔ English translation system, but those will be less relevant for the new language pair. It would be possible to translate a sentence like "When does the flight to Tokyo leave?" using the Spanish ↔ English translation system, but not a more appropriate sentence like "When does the flight to Madrid leave?". This issue is related to both the home country and the visiting country.

A corpus generally contains two kinds of sentences:

- Sentences that are "international" and are equally relevant in all or at least a large number of language pairs/countries
- Sentences that are "country-specific", i.e. only relevant for certain language pairs/countries

Table 6.10 gives some examples of the two sentence types. In this case the country-specific sentences would mainly be relevant in a Japanese context.

International Sentences	Country-specific Sentences
Where is the bathroom?	How can I get to *Kyoto*
When do you serve dinner?	Do you serve *Sushi*?
When does the flight leave?	I would like to visit the *Imperial Palace*.

Table 6.10: International and country-specific sentences

Applying the categorized name lists and tags as proposed in the previous section allows the separation of the general international parts of a sentence from the country-specific named entities (see table 6.11).

International Sentences	Named Entities	
	Japan	Spain
How can I get to @CITY	Kyoto	Madrid
Do you serve @DISH?	Sushi	Paella
I would like to visit the @SIGHT.	Imperial Palace	Alhambra

Table 6.11: International sentences and relevant named entities

At this point all sentences can be translated while keeping the same tag in the translation. It will not be possible to decide on the name category level

if a category should be translated or has to be re-collected. The reason here is that basic and very common entries in most categories will be useful in many languages/countries while very specific entries will only be useful in the specific country.

In dishes for example the omnipresent “pizza”, “hamburger” and “hot dogs” should be included for all countries, potentially also the “sushi” example, while very rare and specialty items should not be included. The same is true for city names where the main cities of the biggest countries should probably be available for all language pairs, while smaller towns will certainly not be relevant.

6.3 Personalizing Translation Models

6.3.1 Motivation

It was shown in the previous section how name coverage in machine translation systems can be naturally improved by collecting large name lists for various categories in the respective domains. These name lists will be useful for any translation system, as they introduce high quality translations for unknown and important words. It could also be shown that even the BLEU scores are increased, which are not always sensitive to these improvements.

Unfortunately these name lists have to be very large to be effective in a real situation. It is just unknown what people might say, or where they would like to go. If a translation system is running on a server, the name lists will usually not negatively affect the system. Most names will have only one or a very small number of possible translations so the translation lattice will not grow heavily. The server just has to keep the name list in memory and apply it if necessary.

A problem arises, however, when the name lists should be put on a small device like a PDA or a cell phone. Here, the size alone might make it impossible to put all the lists on the device. In addition, the approaches presented in chapter 5 will not work here, as each name is potentially important and there will not be a large number of translation alternatives for each name. Even if the device is able to hold the whole name lists, the translation system will usually be part of a speech to speech system. Here the large vocabulary the name lists will introduce could negatively affect the speech recognition performance. More words become potential candidates and different words might have similar or even identical pronunciations, which leads to confusability and an increase in search space (Huangxi et al., 2001).

New words are also continuously added to a language and these words cannot be prematurely captured in these static name lists. It is estimated that about 25,000 new words are added to the English language each year (Kister, 1992).

6.3.2 Background Lexicon

A possible solution to this problem for the translation system could be an online background lexicon running on an internet server. This background lexicon could provide the name translations to the PDA dynamically as needed during the translation process. It would be trivial to add new words to the background lexicon, and they would be immediately available to the translation devices. The PDA would only need to store the general phrase pairs;

the names and specialty vocabulary would be provided via the internet server. As most sentences will only contain a small number of names, it should also not affect the translation speed too heavily.

The disadvantages of this approach are the constant need for an internet connection by the mobile device. It is unlikely that this will be possible at all times in the discussed domains. A tourist might have an internet connection in his hotel or in central areas of cities, but not in rural areas or national parks. It might also be hard to do the speech recognition via a narrow-band online connection.

6.3.3 Dynamic Personalization

The limited internet connection availability makes it difficult to just rely on such a background lexicon, but it is also not possible to store all name lists on the small device. To solve these issues, a personalization method is proposed that will require only limited internet access, but will eventually provide each individual with a specifically personalized translation system.

6.3.3.1 Specific User - Specific Interest

It is true that if all users are viewed as an overall group, they will need a large number of named entities and specialty vocabulary to effectively communicate. This was the main reason to argue that translation systems need a larger vocabulary. On the other hand, each translation system is at one time only used by one user. This user only has a relatively small number of specific interests and will not need all specialty terms relevant for other hobbies or interests. This will differ greatly between different users, so it will not be possible to know beforehand which names will be relevant. The relevant vocabulary could also change during usage as a tourist travels through different areas of a country or changes his interests.

6.3.3.2 Personalized Translation Models

The proposed idea focuses on personalization of translation models that will be introduced and illustrated with an example. At the beginning, translation systems for all users are pre-loaded with the same standard phrase pairs and the most common name lists that will fit on the device. Table 6.12 lists some phrases from the 3 identical phrase tables for 3 users.

User 1	User 2	User 3
Yes	Yes	Yes
No	No	No
Hotel	Hotel	Hotel
Restaurant	Restaurant	Restaurant
Sushi	Sushi	Sushi
Starbucks	Starbucks	Starbucks
...	...	...

Table 6.12: Start situation - Same phrase table for all users

At this point the users are able to translate the common sentences, but the translation will not work for very specific terms. The system will, however, keep detailed statistics of the usage of the available phrase pairs. It will also note the occasions where a term occurred that could not be translated. This would either require the speech recognition part to be able to recognize the word while it is unknown to the translation system, or the user would be required to type in the unrecognized parts of the sentence. Table 6.13 now has additional frequency statistics for each phrase.

User 1		User 2		User 3	
Yes	12	Yes	11	Yes	32
No	10	No	21	No	12
Hotel	0	Hotel	4	Hotel	19
Restaurant	3	Restaurant	8	Restaurant	8
Sushi	2	Sushi	15	Sushi	0
Starbucks	7	Starbucks	0	Starbucks	0
...		...		...	
Unknown		Unknown		Unknown	
McDonald's	2	Sailboat	5	Shoe	7
A-1 Jetfuel	1	Sailing	2	Sole	3
Rotor	1	Keel	1	Mastercard	2
...		...		...	

Table 6.13: Collected statistics and unknown phrases after use

Once the systems are able to connect to the web service, they will send their unknown words and phrases and receive the translations (if available). The system can also send additional phrase pairs that are related to the unknown words. For example, the first user might not only receive the

translations for "A-1 Jetfuel", "McDonald's" and "Rotor" but also some related words and phrases like "Kerosene", "Oil", "Burger King" and "Blade" (see Table 6.14). The systems then integrate the newly received phrase pairs into their phrase tables and are able to translate these words in the future. The systems also remove phrase pairs that were never used.

Phrases requested	Phrases received
A-1 Jetfuel	A-1 Jetfuel
	Kerosene
	Oil
McDonald's	McDonald's
	Burger King
Rotor	Rotor
	Blade
...	...

Table 6.14: Requests to online service might return related phrase pairs

User 1		User 2		User 3	
Yes	12	Yes	11	Yes	32
No	10	No	21	No	12
~~Hotel~~	0	Hotel	4	Hotel	19
Restaurant	3	Restaurant	8	Restaurant	8
Sushi	2	Sushi	15	~~Sushi~~	0
Starbucks	7	~~Starbucks~~	0	~~Starbucks~~	0
...		...		...	

Table 6.15: Removing unused phrase pairs

The overall system could also consider other users' statistics when adding or eliminating phrase pairs. If a large group of users needed a specific phrase, it could already be added to the phrase table of systems where the user did not yet use this unknown phrase. As so many people needed it, it can be estimated that it is a universally used phrase that could be beneficial to all users. Similarly, a phrase should not be eliminated if many other users needed it.

The actual decision to add or eliminate certain phrase pairs should, however, stay with the mobile system, as each system might have different restrictions. A laptop might have much more space and computing power to handle larger phrase tables compared to a cell phone.

For example the policy could be:

- Cell phone:
  Keep actually used unknown words only
- PDA:
  Keep actually used unknown words + related phrase pairs
- Laptop:
  Keep actually used unknown words + related phrases pairs +
  unknown words from other users

This allows this service to be used by a wide variety of translation systems.

6.3.3.3 Improving the Online Service

The fact that the systems will regularly contact the online service allows the service provider to gather overall statistics of phrase pair usage and common unknown words. Analysts will be able to:

- View, check and improve the most commonly used phrase pairs
- Add unknown phrase pairs starting from the ones most frequently requested. This will also allow the identification of topics that might not be covered well enough. The analysts could add additional training data to improve the situation for a topic in high-demand.
- View the most commonly used sentences and improve their translations.

All these improvements can be based on the frequency of the phrase pair/sentence usages, so the analysts can be most effective with the highest impact for the largest number of users. Data privacy issues should be considered, and the anonymity of the users should be guaranteed.

6.3.3.4 Improvements by the User

An additional possibility is to allow the user himself to add unknown words, if he has access to the correct translation. These additions could also be automatically distributed to other users via the online service.

6.3.3.5 Analysis and Evaluation

A complete implementation and evaluation of this personalization method of phrase tables was beyond the scope of the thesis work, but this section will discuss the main issues and give first results.

The initial step is to put a translation system on the devices that has not been personalized yet. For medical users the baseline system from section 6.2.2.4 could fulfill this purpose. It is trained on medical dialogs, but the coverage of specialty terms and named entities is limited.

In the next step one of the users might use the system to translate the 500 lines of medical dialog that was used as the test set. In this case the BLEU score is 0.1912, but the communication success of this user will be heavily limited by the 411 unknown words that occur in this dialog.

The task of the translation system in this first dialog is to note these unknown words. It will be necessary for the user to type them in, if the speech recognition system is not able to recognize them.

Once the system is able to connect to an online service, it can be provided with additional translation pairs from a background lexicon. In this case the medical dictionary extracted from the UMLS with over 5 million translation pairs can serve as this background lexicon. The earlier results show that is does not contain all 411 unknown words, but it does contain 109 of them. It is also possible that it contains more appropriate translations for other phrases that occur in the test data. The online service can now send the translations for the unknown words and also improved translations for other phrases to the system and it can integrate it into its phrase table. This step alone improves the BLEU score on the tested dialog to 0.2015. If the phrases are also integrated into the language model the score further improves to 0.2034.

If a class based framework is used, all generic phrase pairs with class tags can already be stored on the mobile device. It should just be avoided to put all entries in the name lists on the mobile device and those should be provided as needed.

These arguments and the earlier results show that it is possible to update the translation system and realize these benefits while keeping the overall phrase table size small. This method will certainly require further experimentation with non-expert users in various situations.

6.4 Communicating Unknown Words

6.4.1 Motivation

Even with the added name lists and specialty vocabulary, the translation system will encounter unknown words and phrases that are not covered. One of the reasons is that any (living) language is constantly changing. As previously mentioned, English is estimated to add 25,000 words per year. It also seems nearly impossible to cover every word in the English vocabulary of approximately 500,000 to 600,000 (Kister, 1992) with bilingual texts (other languages often have even larger vocabularies). For comparison, the vocabulary of the full English BTEC corpus is less than 15,000 words.

Overall, even with background lexicons and name lists, the translation system still has to handle unknown words. The method that will be presented here has to be seen as a last-resort method that should only be applied after all other sources and methods have been exhausted.

Two example sentences from the later experiments Spanish → English are shown in table 6.16: In both sentences the rest of the sentence apart

Translation:	*revelan* you have diabetes
Reference:	they reveal that you have diabetes
Translation:	i am sure you will be getting a great *mejoría* in 4 or 5 weeks
Reference:	i am sure you will feel a great improvement in 4 to 5 weeks

Table 6.16: Example sentences with unknown words

from the unknown word is translated quite well. However, in both sentences the unknown word contains a lot of information. Especially in the second sentence, as the patient will not know what he has to expect in 4 or 5 weeks. It is not even clear if it will be a positive or a negative event. The first sentence is relatively understandable, but there is the possibility that the unknown word might negate the meaning.

6.4.2 Related Work

Various approaches to translate or generally deal with unknown words have been proposed before. One idea relies on morphological similarities to map the unknown words to known words as used in Mermer et al. (2007). This can give very good results, especially if the source language is morphologically

richer than the target language. Multiple words in the source language that only differ in their inflection will then be aligned to a single word in the target language. Even if not all of them were seen and the morphological similarity can be detected, the correct translation can be produced. A problem can arise if the inflectional difference indicates a different part of speech, as the approach will then produce an incorrect part of speech in the target language.

This approach is comparable to the proportional analogies in Lepage and Denoual (2005). The technique of Analogical Learning was also later directly applied to the translation of unknown words in Langlais and Patry (2007). The idea is to find the same relationships between the word in the source language for which the translation is not known and another word where the translation is known, as well as the same relationships for other word pairs for which both translations are known. An example translating French to English (from Langlais and Patry (2007)) is the unknown word "futilité". It has the same "relationship" to the known word "futilités" as the known pairs "activité:activités" and "hostilité:hostilitiés". A seed lexicon gives the translations and the known pairs are compared as well. In this case, "action:actions" and "hostilities:hostility". This gives two different ways how to form the translation of "futilité" given the translations for "futilités". The first way is applied to the possible translation "gimmicks" generating "gimmick", on the second possible translation "trivialities" the second way is applied forming "triviality" which are both correct translations for "futilité".

Other teams investigated comparable corpora as a source for unknown word translations (Fung and Yee, 1998; Rapp, 1999; Takaaki and Matsuo, 1999). In these cases, a seed lexicon is used to translate parts of the sentences for one side of the corpora. If contexts or even whole sentences are very similar based on the seed translations, the remaining words are translation candidates for each other. In the publication Haghighi et al. (2008) a similar approach is proposed under a *canonical correlation analysis* (CCA) framework.

Other approaches mainly target named entities, as here special transliteration rules are often applied that try to reproduce the original phonemes using the phonemes (and graphemes) of the target language. This is mainly applicable if the character sets of source and target languages are different[2]. Techniques are described in Zhao et al. (2007) and Huang (2005, 2006). Generally, transliteration models are introduced that learn character and character group transliteration from given examples and can produce a transliteration hypothesis. It can be beneficial to spell-check this hypothesis against a large vocabulary to correct additional transliteration errors.

[2]The task is much easier, often trivial, if the character sets are identical.

A data mining approach presented in Zhang et al. (2005) and Huang et al. (2005) uses web queries to find web pages that contain both the source and target named entity. This uses the fact that key phrases like movie titles are often given in the translated and original form next to each other, particularly in Chinese text.

More closely related to the approach presented here are the methods discussed in Callison-Burch et al. (2006) and Cohn and Lapata (2007). The idea is to generate paraphrases for unknown words and phrases from parallel bilingual texts. These paraphrases retain the same meaning, but can be translated. Additional bilingual corpora are necessary, but they can have another second language. This means the English paraphrases can be extracted from an English/German corpus and later applied to an English → Spanish translation task.

It should be pointed out, that the presented approach is related to these approaches. It should be seen as an additional step to deal with remaining unknown words after these approaches and the methods discussed in the earlier chapters are applied. Especially approaches based on morphological similarities have shown very good results, but they can only be applied if reasonably close words are in the translation lexicon. The other techniques also have an inherent limitation, e.g. the transliteration approaches are only applicable to named entities or need additional bilingual (or at least comparable) corpora. This technique and the results were presented in Eck et al. (2008).

6.4.3 Communicating Unknown Words

Everyone who is starting to learn a foreign language and even people with more experience will find themselves in situations where they cannot recall the correct translation for a specific term. They might also not be able to think of an alternative term or synonym. People in these situations usually resort to explaining the missing word using words and phrases they know. It can even happen in a native language where people might not recall a specific term and explain it instead. Someone might want to translate “annually” to Spanish, but does not recall the correct term (“anualmente”). However, he might be able to say “once a year” or “one time per year” in Spanish as those words are more common and potentially easier to translate for a non-native speaker.

In order to use this approach in a translation system, these “explanations” or “definitions” have to be automatically generated for unknown words. This can be accomplished using monolingual lexicons or encyclopedias on the source language side.

These monolingual sources offer a much better word coverage as they actually try to cover every word in a language. The Oxford English Dictionary (Simpson and Weiner, 1989), for example, contains 616,500 word forms in its latest edition (also available online at `www.oed.com`).

Once the "definition" has been extracted, it has to be translated using the automatic translation system. However, the definition could contain additional unknown words. Lexicons usually try to limit the vocabulary used in the definitions, as even a regular human user might not know some specific terms. An example here is the *Oxford 3000*, a list of words that is the basis for the "Oxford Advanced Learners Dictionary". Definitions should only contain words from this list of 3000 basic words. The dictionary does not, however, completely follow this demand (Wehmeier, 2007). Other knowledge sources might also try to prefer simple words in their explanations, but that cannot generally be guaranteed.

6.4.4 Alternative Approach

An alternative approach would be to use semantic lexicons (e.g. WordNet for English, `wordnet.princeton.edu`) that can give synonyms for entered terms. The English WordNet for example produces "yearly" and "per annum" for the example term "annually". However, WordNet type databases are not available for a large number of languages, and their coverage is also more limited as preliminary tests showed. From this perspective, the approach using dictionaries and encyclopedias seemed more flexible, particularly as dictionaries also occasionally list synonyms.

6.4.5 Process Overview

Figure 6.1 shows the flowchart for the proposed process. Starting from a baseline translation, the remaining unknown words are identified, and a definition is generated for each word. The definition is translated using the automatic translation system and inserted in the baseline translation to form the final hypothesis.

6.4.5.1 Extract Definition for Unknown Word

Given an unknown word, the first step is to find the definition for the unknown word in the source language. The main goals for a definition of an unknown word are that it is unambiguous, short and as easy to understand as the original word. Another constraint is that it should not use any words that

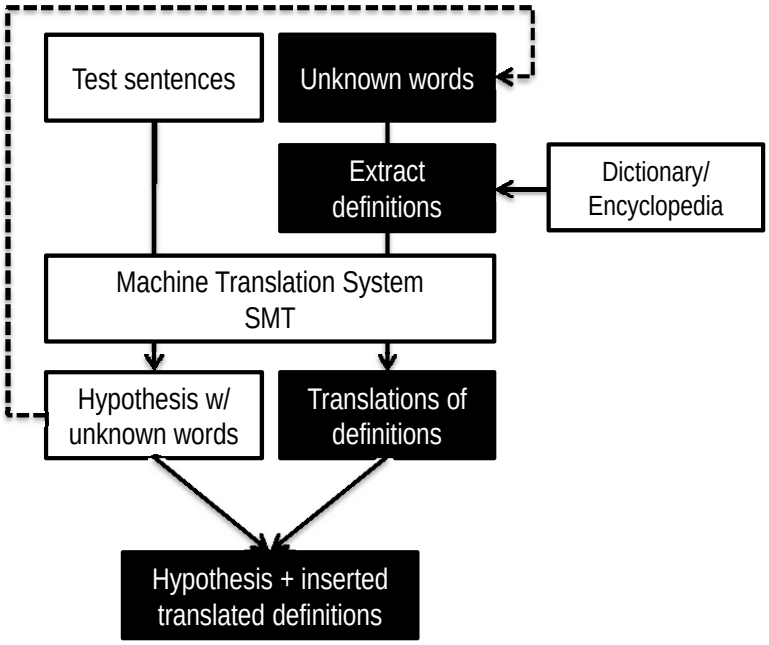


Figure 6.1: Process overview - Handling unknown words

are unknown to the translation system, as it would again not be possible to translate the complete definition. Concerning the source for the definitions, two main cases were identified, depending on the type of unknown word:

- Regular content words (that are not named entities): Workforce, thrice, ascertain, annually, biweekly, ancestry

- Named entities: Cairo, Lima, Pennsylvania, McDonald's, BMW

Dictionaries For regular content words that are not named entities, standard online dictionaries usually provide concise and short explanations. For the following experiments, dictionary.com (`dictionary.reference.com`) and wordreference.com (`www.wordreference.com`) were used as sources for English and Spanish definitions respectively. Both dictionaries list multiple meanings per word. If necessary, a word sense disambiguation component could help to find the best fitting one at this stage.

Explanations in a dictionary usually fall into one of three categories:

- Synonym: (e.g. *occur*: to happen)
- longer explanation: (e.g. *workforce*: the total number of workers in a specific undertaking).
- example sentence: (e.g. *determine*: Demand for a product usually determines supply)

Figure 6.2 shows the results from dictionary.com for the search term "ancestry".

Dictionary.com Unabridged (v 1.1) – *Cite This Source* – *Share This*
an·ces·try [**an**-ses-tree *or, especially Brit.*, -*suh*-stree]
Pronunciation Key - Show IPA Pronunciation
–noun, plural **–tries.**
1. family or ancestral descent; lineage.
2. honorable or distinguished descent: *famous by title and ancestry.*
3. a series of ancestors: *His ancestry settled Utah.*
4. the inception or origin of a phenomenon, object, idea, or style.
5. the history or developmental process of a phenomenon, object, idea, or style.

[Origin: 1300–50; ME, equiv. to *ancestre* ANCESTOR + –Y^3; r. ME *aunce(s)trie* < AF.]

—Synonyms 1. pedigree, genealogy, stock. 3. family, line.

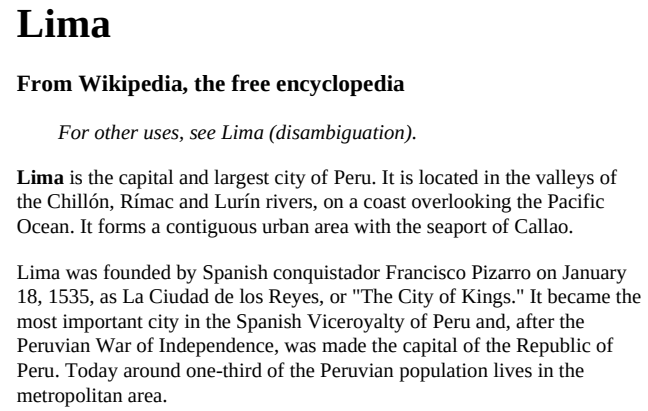
Dictionary.com Unabridged (v 1.1)
Based on the Random House Unabridged Dictionary, © Random House, Inc. 2006.

Figure 6.2: Dictionary.com results for search term "ancestry" (excerpt)

Wikipedia Named entities are a special case, as regular dictionaries do not contain many named entities or specific brand names. An encyclopedia offers better coverage here. For these experiments, the automatic extraction from Wikipedia articles was investigated. Wikipedia is a free online encyclopedia that can be edited by any user. As of January 2008, it is available in 256 languages with more than 10,000 articles in 75 of those languages. The English section contains the most articles with over 2 million. There have been concerns and discussions about the reliability and the objectivity of Wikipedia articles (see e.g. Giles (2005), Cohen (2007)), but the effect on translation should be minimal.

A big advantage of Wikipedia is the fast addition of new terms as every user can edit and add articles so the encyclopedia stays very up-to-date, especially on popular topics.

Wikipedia articles can be extremely long and it is not useful to translate a long article just to communicate a single unknown word. However, it was empirically found that the first sentence of the Wikipedia article usually gives a good definition of the term if the term can be clearly defined.



Lima

From Wikipedia, the free encyclopedia

For other uses, see Lima (disambiguation).

Lima is the capital and largest city of Peru. It is located in the valleys of the Chillón, Rímac and Lurín rivers, on a coast overlooking the Pacific Ocean. It forms a contiguous urban area with the seaport of Callao.

Lima was founded by Spanish conquistador Francisco Pizarro on January 18, 1535, as La Ciudad de los Reyes, or "The City of Kings." It became the most important city in the Spanish Viceroyalty of Peru and, after the Peruvian War of Independence, was made the capital of the Republic of Peru. Today around one-third of the Peruvian population lives in the metropolitan area.

Figure 6.3: Wikipedia result for search term “Lima” (excerpt from main entry)

Figure 6.3 shows the first part of the main Wikipedia entry for the term “Lima”. The main definition “...is the capital and largest city of Peru” is there. It would, however, be necessary for the dialog partner to have the knowledge to be able to deduce the actual term from just this definition.

Homonymy and Polysemy Many words have multiple meanings either through polysemy or homonymy. Polysemous word meanings are related so they will likely not vary too much, and it will have little influence regardless of which definition is actually chosen. On the other hand homonymous word meanings could be very different (Cruse, 2004). A word sense disambiguation component could be valuable here. This was not investigated, as the general impression is that word meanings of the unknown words do not vary heavily and are usually closely related. Therefore, the expectation is that such a component would not significantly increase the performance. The first and third result for the term “ancestry” shown in figure 6.2 are closely related with the second result only slightly different. The last two definitions define meanings that are more abstract, but are still related, as all these meanings are polysemous. This general claim is supported by Twilley et al. (1994). In this article 44% of all English words tested were ambiguous, but 85% of frequent English words. This means that words with a lower frequency

are less ambiguous, and, naturally, unknown words will fall into the lower frequency category.

6.4.5.2 Translation of the Definition

It would also not be valuable to find a definition for the correct meaning if this definition contains a large number of unknown words and cannot be translated. Named entity explanations especially tend to use further named entities that might also not be known to the translation system. However, as pointed out, lexicons try to limit the number of words that are used within the definitions, but this cannot be assumed for Wikipedia articles.

For this reason, the definition with the fewest number of unknown words was chosen as well as the first definition for a comparison. The first definition has the advantage of likely being the most common meaning.

Another option would be to ask the user for input. It would be simple to show a number of possible definitions and have the user select the correct one. This also makes him aware that there was an unknown word and allows him to rephrase the sentence if the definitions do not meet his needs.

Furthermore, the style of the definitions can be very different from the domain of the actual translation system. In this example the translation system was trained on tourism phrases, a completely different style than the short and concise definitions. Definitions are also frequently grammatically incomplete sentences which adds to this problem.

6.4.5.3 Insert Translated Definition into Original Hypothesis

To finally produce the improved translation, the translated definition has to be introduced into the baseline translation. Just replacing the unknown word with the definition is questionable as this might make the sentence unclear and confusing in the target language. The experiments also show that the definition does not always describe the word in the same part of speech, as dictionaries usually project words to their base forms.

For these reasons it is valuable to clearly mark the definition as such and leave the decision of wether that definition defines a word or a short phrase to the speaker of the target language. This way, affecting the coherence of the rest of the sentence is avoided.

Table 6.17 shows the two previous example sentences with translated and inserted definitions. The unknown word is marked by “UNK” and the translated definition is added. Both sentences clearly improve with the added definitions. It is assumed here that the output is text as it was not yet investigated how this could be optimally integrated into speech output.

Improved hypothesis:	(UNK: revelan: undiscovered it secret) you have diabetes
Translation:	*revelan* you have diabetes.
Reference:	they reveal that you have diabetes.
Improved hypothesis:	i am sure you will be getting a great (UNK: mejoría: getting better) in 4 or 5 weeks
Translation:	i am sure you will be getting a great *mejoría* in 4 or 5 weeks
Reference:	i am sure you will feel a great improvement in 4 to 5 weeks

Table 6.17: Sentences with inserted and translated definitions compared to baseline and reference

6.4.6 Experimental Results

6.4.6.1 Monolingual Experiment

The first experiment was intended to find out for how many monolingual English words a meaningful definition could be extracted. For this experiment, the 16 English reference translations of the IWSLT 2004 test set (Akiba et al., 2004) were chosen, and all unknown words were determined compared with the English Full BTEC corpus. Overall, 236 words out of the references are unseen and definitions for those were extracted automatically from Dictionary.com (`www.dictionary.com`). In this case the first definition was always chosen. One human subject (native English speaker) judged the adequacy of the extracted definitions on a scale of 1(worst) to 5(best) (compare Fordyce (2007), Paul (2006)). The subject was asked to judge the adequacy "as if the definitions were translations". Missing any contexts, the evaluator was also instructed to assume the most common meaning for each word.

Figure 6.4 shows the distribution of the different adequacy scores that were assigned. For 46 words, no definition could be extracted and the worst score of 1 was assigned. These words include typographical errors ("1've"), exclamations ("Yum") and certain slang terms. 9 other words also received the worst score, mainly due to definitions for unusual word meanings. 88 words overall received a score of 2 or 3. This was mainly due to incorrect conjugations or referring to an incorrect part of speech e.g.:

- *summoning*: To call together; to convene
- *locations*: The act or process of locating

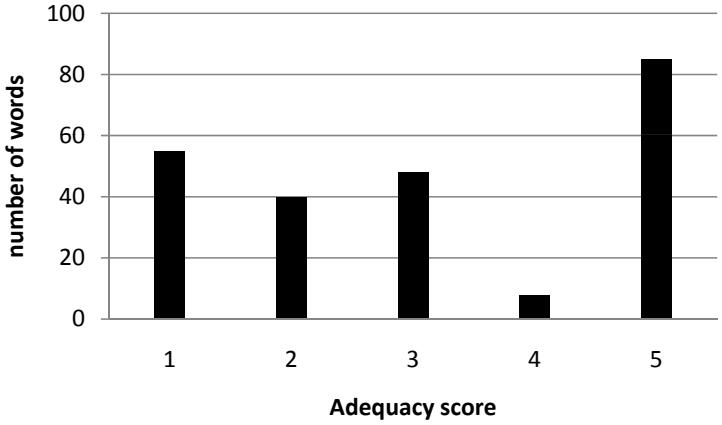


Figure 6.4: Monolingual adequacy scores

Dictionary.com and other dictionaries project inflected word forms to their base form, which leads to these issues. It is not clear how severe this issue will be in actual usage and the relatively low score might not always be justifiable here. 8 definitions received an adequacy score of 4, and 85 times the best score of 5 was assigned. The very low number with adequacy 4 is understandable as this means that the definition is only slightly incorrect and this was not expected in this setup. An adequacy of 4 is more common for automatic translation with minor errors, but these definitions were manually created for the dictionary. The overall average score was 3.12.

6.4.6.2 Bilingual Experiments

The actual question is, however, if it is possible to extract these definitions in a source language and translate the definitions to a target language while conserving the complete meaning of the extracted definition. Standard automatic evaluations will not be able to show the improvements targeted here, so again a subjective evaluation was performed.

Experimental Setup For training data, an English-Spanish tourism phrase corpus was used (BTEC). A 500 line test set was used consisting of medical dialogs to test the approach translating from Spanish to English. This is the same test set as was used in section 6.2.2. The number of unknown words is large enough at 289 to allow meaningful experiments.

The translation system used for the baseline translations and also the translation of the extracted definitions is a standard statistical machine translation system using an online phrase extraction method (PESA) and

a 6-gram language model trained on the English part of the bilingual training corpus (Vogel, 2003, 2005; Eck et al., 2006) (see table 6.18).

Bilingual Training Data			Monolingual Training Data	
	Spanish	English		English
Lines	123,416	123,416	Lines	123,416
Words	852,362	903,525	Words	903,525
Translation Models	PESA			
Language Models	SuffixArray 6-gram			
Test Data	500 lines, medical dialogs			
Baseline Adequacy	2.00 (please note explanations)			

Table 6.18: Experimental setup Spanish → English

Extracting Definitions This test set contains 289 unknown words in Spanish for which Spanish definitions from `www.wordreference.com` were extracted. For the initial experiment, the first definition was chosen as it is likely the most common meaning. In a second experiment, the definition with the lowest number of unknown words was chosen. The argument for this is simply that the definition has to be translated. For 86 words no definition could be extracted. As in the previous experiment, these words are mainly typos, named entities and brand names that are not available in `www.wordreference.com`. For the remaining 203 words definitions were extracted. Table 6.19 compares how often definitions with 0 to 2 and more unknown words could be extracted in both approaches.

Unknown words	First definition	Lowest number of unknown words
0	33	57
1	46	61
2	49	46
> 2	75	39
Average/definition	2.50	1.71

Table 6.19: Unknown words in extracted definitions

The definitions contained, on average, 2.50 unknown words if the first definition was extracted and 1.71 unknown words if the definition with the lowest number of unknown words was chosen.

The definitions were again subjectively judged for adequacy according to the scale in table 6.20. Here the translations were inserted in the respective hypothesis sentences as described in section 6.4.5.3.

1	Worse than unknown word, misleading
2	No change compared to unknown word
3	Clear improvement
4	Good translation
5	Perfect translation

Table 6.20: Adequacy judgments for bilingual experiments

A score of 1 was assigned if the translation became actually misleading and was clearly worse than the unknown word. This means the sentence had to make reasonable sense but was also misleading. A score of 2 was assigned if the inserted definition did not give any benefit over the unknown word. This also implies that the baseline score with all unknown words would be a score of 2. Scores 3 to 5 were assigned for improvements compared to the original sentence. Figure 6.5 illustrates the adequacy results.

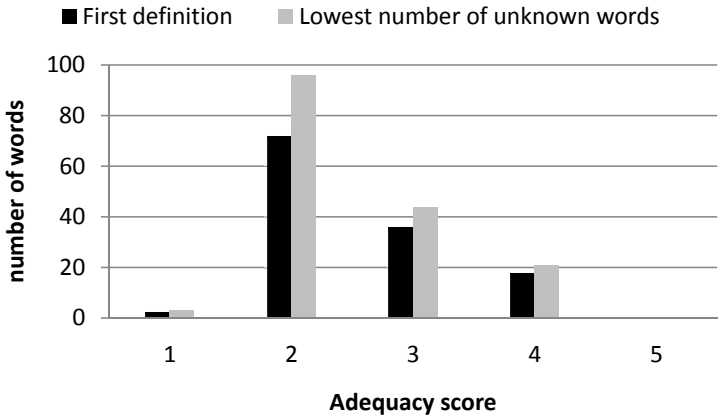


Figure 6.5: Adequacy scores in the bilingual experiment

Generally, if more than 2 unknown words are in the definition, the translations were not understandable and received a score of 2, which leaves 128 first definitions and 164 definitions with the lowest number of unknown words. The average adequacy score for those definitions is 2.55 for the first definitions and 2.51 for the definitions with the lowest number of unknown words. This is only slightly lower than the score for the first definition, but it produced definitions for more words. A score of 1 was only assigned two times

and three times respectively while a score of 5 (perfect translation) was never assigned. If only the definitions with 0 unknown words are considered, the averages are 2.96 for the first definition (33 instances) and 2.67 (57 instances) for the definition with the lowest number of unknown words.

6.4.6.3 Translation Examples

Table 6.21 lists some examples for translated definitions with no unknown words in the first definition (so it was also the definition with the lowest number of unknown words). Most examples show clear improvements.

Spanish word:	innecesario
Translated definition:	is not it necessary
Reference:	unnecessary
Spanish word:	repetitivos
Translated definition:	to repeat
Reference:	repeat
Spanish word:	acidos
Translated definition:	you have taste sour
Reference:	acidic
Spanish word:	energías
Translated definition:	power be able
Reference:	energy
Spanish word:	intranquilas
Translated definition:	eager nervous
Reference:	stressful

Table 6.21: Translation examples - Definitions without unknown words

Table 6.22 compares the translations of the first definition with the translations of the definition with the lowest number of unknown words. This clearly illustrates the correlation between the number of unknown words and the quality of the translation. The last example shows one of the instances where the translated definition was judged worse than the unknown word. “Radiante” is translated correctly, but in this context describing “radiating pain” the incorrect and misleading meaning was chosen.

Spanish word:	brota
First definition:	quite UNK the floor the UNK
Fewest unknown:	get out to the surface disease
Reference:	outbreak (disease)
Spanish word:	dolían
First definition:	quite UNK pain in a part of the body
Fewest unknown:	causing consumers' pain
Reference:	sore
Spanish word:	asquerosa
First definition:	4x UNK
Fewest unknown:	disgusting to have UNK
Reference:	disgusting
Spanish word:	radiante
First definition:	UNK bright
Fewest unknown:	very happy, or satisfied for something
Reference:	radiating (pain)

Table 6.22: Translation examples - Definitions with unknown words

6.4.6.4 Extracting Definitions from Wikipedia

To extract definitions for named entities, Wikipedia was used in preliminary experiments. Bilingual experiments could not be done due to the lack of an appropriate test set rich with named entities. However, it is reasonable to assume that results similar to before can be achieved if concise definitions for the unknown words are available. This is also the main issue in Wikipedia, as articles tend to be very long and not concise. It was empirically found that the first sentence of an article tends to give a good definition, if a short definition is possible. Table 6.23 shows some examples.

Unknown word	First Wikipedia sentence
Lima	is the capital and largest city of Peru.
Kilimanjaro	is an inactive stratovolcano in north-eastern Tanzania.
Tempura	is a classic Japanese dish of deep fried lightly-battered vegetables or seafood.
Bolivia	is a landlocked country in South America.

Table 6.23: Example definitions extracted from Wikipedia

It is clear that the dialog partner has to have additional world knowledge to understand what is being defined to get to the actual term. However, there could also be situations where the more general term e.g. “city” or “city in Peru” for “Lima” could be better than not having any translations. The last example shows an instance where the definition is not unambiguous (Bolivia is not the only landlocked country in South America). This definition is also an example where a significant amount of general knowledge would be necessary.

6.4.7 Analysis

The experiments show that the proposed approach can give considerable improvements in communicating unknown words. The main limiting issues are remaining unknown words in the extracted definitions and the projection of inflected words to a base form, which can lead to differences concerning the part of speech. It could be shown that selecting the definition with the lowest number of unknown words can improve this situation while the translation quality still improves.

It might be valuable to develop specialized translation systems to translate the definitions as the domain mismatch in the experiments clearly influenced the translations. Further experiments with definitions for named entities extracted from Wikipedia articles will be necessary. It might also be valuable to investigate summarization approaches to improve the extraction of concise and unambiguous definitions from the long articles.

The question how this can be included in a complete speech to speech translation system remains as well. It will most likely be necessary to type in the unknown word, as it cannot be assumed that it is part of the speech recognition vocabulary. At that point the user could also be asked to select the most fitting definition from a number of presented options.

Chapter 7

Summary

The preceding chapters introduced various approaches to develop deployable spoken language translation systems given limited resources. Three main factors were investigated:

- Low cost portability for fast transfer to new language pairs
- Translation models for small, mobile devices
- Improving named entity and specialized vocabulary coverage

7.1 Low Cost Language Portability

To cover new language pairs effectively, sentence sorting schemes were introduced that order the source sentences of a given monolingual corpus depending on their estimated importance. Translating the top n sentences of the sorted corpora results in far better bilingual corpora than leaving the sentences in random or original order. The main benefit is realized by automatically identifying word and phrase repetitions. An effort is then made to limit these repetitions in the corpus given to the human translators. The most successful algorithms are coverage based, with the goal of producing a corpus that contains each word or n-gram at least once. It is possible to prove that these problems are NP hard or NP complete. This depends on the individual problem definition, but efficient approximation algorithms can be applied.

In the static sorting approaches, only the monolingual source corpus is considered, while the dynamic sorting approaches take the previous translations into account as well. The dynamic approaches are able to identify and separate beneficial and non-beneficial repetitions. This allows them to

slightly outperform the static approaches on certain tasks and data sizes. On the other hand, the computational and organizational complexity increases heavily, which will limit the practical applicability of the dynamic approaches.

7.2 Models for Mobile Devices

The main issues in porting a translation system to a lightweight and mobile device are the computing power and the memory requirements of the translation and language models. The proposed approach to decrease the size of the translation model tries to estimate the probability of a certain phrase pair being used in a translation hypothesis. This is accomplished by translating a large amount of text and collecting usage statistics for each phrase pair. This allows for smaller translation models than any previously known standard approach is able to produce.

It proved valuable to incorporate the top 10 entries of the N-best list to get statistics for more phrase pairs. It was not possible to significantly improve the results using any metric information. This might have been partly due to specific circumstances in the experiment; specifically, the fact that the model-best and metric-best paths were often similar.

7.3 Improving Vocabulary Coverage

Increasing the coverage of names and specialty vocabulary is, at first, mainly a manual (or at least semi-manual) task. While names can often be successfully transliterated, this is not generally possible and rare for specialty vocabulary. If transliteration is not possible, name lists will have to be manually collected. Some lists could be extracted from websites, databases or other documents, but a manual checking is always advisable.

If large name lists are available, as is the case with the UMLS database, they can be effectively used in a class based framework. This was demonstrated using the cascaded application of phrase tables.

Replacing a name with an automatically extracted definition and translating this definition was also proposed. Experiments showed that this has the ability to improve the subjective translation quality.

Unfortunately, large name lists are problematic for a mobile device, as there might not be enough memory available to store them in full. The proposed solution is based on the fact that each individual user will only need a small fraction of all available names. Systems should be able to

collect unknown words and phrases and update themselves via an internet-based service. This will allow each user to eventually have an updated system that is exactly tailored to his or her interests, needs or specialties.

7.4 Future Work

The main goal of future work should be to find improved ways to evaluate the proposed new approaches. The standard automatic evaluation metrics, like BLEU, that were primarily used in this thesis allowed for a fast experimental turnaround and easy tests of new approaches. They are also the commonly accepted metrics for machine translation evaluation in the research community.

However, a BLEU score has no real meaning for a non-expert user. Even experts might have problems to define what a certain improvement in BLEU score really means to a user. BLEU scores have been shown to correlate reasonably well with human judgments on the individual test sets, but they cannot measure any semantic factors, communication success or task completion rates. A human user will not necessarily care for small translation errors if all the important concepts are transferred to the dialog partner and the communication is successful.

Users would probably prefer a more task oriented semantic measurement of translation performance. Tasks in the tourism domain could be "Booking of flights", "Ordering in a restaurant" or "Asking for directions". The current technology is not able to guarantee that any of these tasks will work in all possible situations. This takes the aforementioned goal of communication success a step further, as multiple communication successes have to happen to complete a task. A translation system with a larger number of named entities might now be able to translate "I would like to book a flight to Lima" correctly. For this single utterance, communication success was achieved. However, the task of "Booking a flight to Lima" will contain a number of turns by the tourist and travel agent, and if one of the turns fails, the whole task completion could be in jeopardy.

This has some similarity to the evaluation methods proposed in Voss and Tate (2006) and Jones et al. (2007). In both publications, the subjective evaluators are asked to answer questions based on the information in the translated sentences. This is more relevant to news wire texts, and a different approach might be necessary in the domains discussed here.

Changing the evaluation procedure to a semantic or more task-oriented method would specifically affect the low cost language portability and the pruned models for small devices. In both cases, the goal is no longer to

match or get close to a baseline BLEU score, but instead to match the task completion rates. At what point can the same tasks be accomplished that were possible to accomplish in another language pair? How much can the model be pruned while not affecting the task completion rate?

Some situations in military and medical applications can be very serious, and translation errors can lead to severe misunderstandings. Standard statistical machine translation systems do not guarantee a correct translation and could be problematic in these cases. It could be beneficial to see if it is possible to augment a statistical machine translation system with a possibility to better handle these cases. This could be done using a combination of detailed confirmation questions, back-translations and confidence estimations, so the user can be assured that his utterance is translated correctly. The MedSLT system for medical translations (Rayner et al., 2006) has the same goal, but uses a transfer-based interlingua approach that is not as flexible as a statistical translation system. The MedSLT system only allows translations in one direction with yes/no replies.

Concerning name coverage, it might be interesting to look at what the actual user expectation is concerning which names can be translated. Which names does a user expect to be covered, and which names will a user accept to not be covered?

The proposed approach to dynamically update the phrase tables of mobile devices based on user needs should also be applied and evaluated, which could not be fully accomplished during the thesis work. This will require a larger scale deployment of translation devices and the setup of an internet-based service to update the translation systems. This is only relevant if multiple non-expert users frequently employ translation systems in real-life situations. There should also be a team of language experts available to review the gathered data and update the background phrase table.

Future work should also consider factors relating to the speech recognition and text synthesis component. Similar work has been done in both areas, but the ultimate goal should be a combined approach. How good can a complete speech to speech translation system be after 1 day, 1 week or 1 month of work by available native speakers, and how can their expertise be used most effectively? Here the limited resources have to be optimally assigned to the three main areas and to specific tasks within these areas. A related question arises if a baseline speech to speech translation system is already available, but has to be improved. Which tasks, that a native human can do, have the most impact on the system performance?

Bibliography

Yasuhiro Akiba, Marcello Federico, Noriko Kando, Hiromi Nakaiwa, Michael Paul, and Jun'ichi Tsujii. *Overview of the IWSLT 2004 Evaluation Campaign.* In Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2004). Kyoto, Japan, 2004.

Noga Alon, Dana Moshkovitz, and Muli Safra. *Algorithmic construction of sets for k-restrictions. ACM Transactions on Algorithms (TALG)*, 2(2):153–177, 2006.

Alison Alvarez, Lori Levin, Robert Frederking, Simon Fung, Donna Gates, and Jeff Good. *The MILE corpus for less commonly taught languages.* In Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006). New York, NY, USA, 2006.

Bogdan Babych, Anthony Hartley, and Serge Sharoff. *Translating from under-resourced languages: comparing direct transfer against pivot translation.* In Proceedings of MT Summit XI. Copenhagen, Denmark, 2007.

Nguyen Bach, Matthias Eck, Paisarn Charoenpornsawat, Thilo Köhler, Sebastian Stüker, ThuyLinh Nguyen, Roger Hsiao, Alex Waibel, Stephan Vogel, Tanja Schultz, and Alan W. Black. *The CMU TransTac 2007 Eyes-free and Hands-free Two-way Speech-to-Speech Translation System.* In Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2007). Trento, Italy, 2007.

Satanjeev Banerjee and Alon Lavie. *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments.* In Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the Annual Meeting of the Association of Computational Linguistics (ACL 2005). Ann Arbor, MI, USA, 2005.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús

Tomás, Enrique Vidal, and Juan-Miguel Vilar. *Statistical Approaches to Computer-Assisted Translation. Computational Linguistics*, to appear, 2008.

Alan W. Black, Ralf D. Brown, Robert Frederking, Kevin Lenzo, John Moody, Alexander Rudnicky, Rita Singh, and Eric Steinbrecher. *Tongues: Rapid Development of Speech to Speech Translation Systems.* In Proceedings of the Human Language Technology Conference (HLT 2002). San Diego, CA, USA, 2002.

Burton H. Bloom. *Space/time trade-offs in hash coding with allowable errors. Communications of the ACM*, 13(7):422–426, 1970.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. *A Statistical Approach to Machine Translation. Computational Linguistics*, 16(2):79–85, 1990.

Peter F. Brown, Stephen Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. *The Mathematic of Statistical Machine Translation: Parameter Estimation. Computational Linguistics*, 19(2):263–311, 1993.

Ralf Brown. *Automated dictionary extraction for knowledge-free example-based translation.* In Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 1997). Santa Fe, NM, USA, 1997.

Allen C. Browne, Guy Divita, Alan R. Aronson, and Alexa T. McGray. *UMLS Language and Vocabulary Tools.* In Proceedings of the American Medical Informatics Association Symposium (AMIA 2003). Washington, DC, USA, 2003.

Chris Callison-Burch, Colin Bannard, and Josh Schroeder. *Scaling Phrase-Based Statistical Machine Translation to Larger Corpora and Longer Phrases.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2005). Ann Arbor, MI, USA, 2005.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. *Improved statistical machine translation using paraphrases.* In Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006). New York, NY, USA, 2006.

Stanley F. Chen and Joshua Goodman. *An empirical study of smoothing techniques for language modeling.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 1996). Morristown, NJ, USA, 1996.

Colin Cherry. *Cohesive Phrase-based Decoding for Statistical Machine Translation.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2008). Columbus, OH, USA, 2008.

David Chiang. *A hierarchical phrase-based model for statistical machine translation.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2005). Ann Arbor, MI, USA, 2005.

Jonathan H. Clark, Robert Frederking, and Lori Levin. *Toward Active Learning in Data Selection: Automatic Discovery of Language Features During Elicitation.* In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2008). Marrakech, Morocco, 2008.

Noam Cohen. *Wikipedia on an academic hit list. NY Times News Service (February 27th, 2007)*, 2007.

David A. Cohn, Zoubin Ghahramani, and Michael I Jordan. *Active Learning with Statistical Models. Journal of Artificial Intelligence Research*, 4:129–145, 1996.

Trevor Cohn and Mirella Lapata. *Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2007). Prague, Czech Republic, 2007.

Michael Collins and Yoram Singer. *Unsupervised models for named entity classification.* In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora. College Park, MD, USA, 1999.

Alan Cruse. *Meaning in Language: An Introduction to Semantics and Pragmatics.* Oxford University Press, 2nd edition, 2004.

Yann Le Cun, John S. Denker, and Sara A. Solla. *Optimal Brain Damage.* In D. S. Touretzky, editor, Advances in Neural Information Processing Systems II. Morgan Kauffman, San Mateo, CA, USA, 1990.

George Doddington. *Automatic Evaluation of Machine Translation Quality using n-Gram Co-occurrence Statistics.* Technical report, National Institute of Standards and Technology (NIST), 2001.

Doug Downey, Matthew Broadhead, and Oren Etzioni. *Locating Complex Named Entities in Web Text.* In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2007). Hyderabad, India, 2007.

Matthias Eck and Chiori Hori. *Overview of the IWSLT 2005 Evaluation Campaign.* In Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2005). Pittsburgh, PA, USA, 2005.

Matthias Eck, Ian Lane, Nguyen Bach, Sanjika Hewavitharana, Muntsin Kolss, Bing Zhao, Almut Silja Hildebrand, Stephan Vogel, and Alex Waibel. *The UKA/CMU Statistical Machine Translation System for IWSLT 2006.* In Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2006). Kyoto, Japan, 2006.

Matthias Eck, Stephan Vogel, and Alex Waibel. *Improving Statistical Machine Translation in the Medical Domain using the Unified Medical Language System.* In Proceedings of the International Conference on Computational Linguistics (COLING 2004). Geneva, Switzerland, 2004.

Matthias Eck, Stephan Vogel, and Alex Waibel. *Low Cost Portability for Statistical Machine Translation based on N-gram Coverage.* In Proceedings of MT Summit X. Phuket, Thailand, 2005a.

Matthias Eck, Stephan Vogel, and Alex Waibel. *Low Cost Portability for Statistical Machine Translation based on N-gram Frequency and TF-IDF.* In Proceedings of the International Workshop for Spoken Language Translation (IWSLT 2005). Pittsburgh, PA, USA, 2005b.

Matthias Eck, Stephan Vogel, and Alex Waibel. *Estimating Phrase Pair Relevance for Translation Model Pruning.* In Proceedings of MT Summit XI. Copenhagen, Denmark, 2007a.

Matthias Eck, Stephan Vogel, and Alex Waibel. *Translation Model Pruning via Usage Statistics for Statistical Machine Translation.* In Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007). Rochester, NY, USA, 2007b.

Matthias Eck, Stephan Vogel, and Alex Waibel. *Communicating Unknown Words in Machine Translation.* In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2008). Marrakech, Morocco, 2008.

Jakob Elming. *Syntactic reordering integrated with phrase-based SMT*. In Proceedings of the International Conference on Computational Linguistics (COLING 2008). Manchester, UK, 2008.

Herman Engelbrecht and Tanja Schultz. *Rapid Development of an Afrikaans-English Speech-to-Speech Translator*. In Proceedings of the International Workshop for Spoken Language Translation (IWSLT 2005). Pittsburgh, PA, USA, 2005.

Uriel Feige. *A Threshold of ln n for Approximating Set Cover*. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

Glenn Flores, M. Barton Laws, Sandra J. Mayo, Barry Zuckerman, Milagros Abreu, Leonardo Medina, and Eric J. Hardt. *Errors in medical interpretation and their potential clinical consequences in pediatric encounters*. *Pediatrics*, 111(1):6–14, 2003.

Cameron Shaw Fordyce. *Overview of the IWSLT 2007 Evaluation Campaign*. In Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2007). Trento, Italy, 2007.

Carol Friedman, Hongfang Liu, Lyuda Shagina, Stephen Johnson, and George Hripcsak. *Evaluating the UMLS as a Source of Lexical Knowledge for Medical Language Processing*. In Proceedings of the American Medical Informatics Association Symposium (AMIA 2001). Washington, DC, USA, 2001.

Pascale Fung and Lo Yuen Yee. *An IR approach for translating new words from nonparallel, comparable texts*. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 1998). San Francisco, CA, USA, 1998.

Jim Giles. *Internet encyclopaedias go head to head*. *Nature*, 438:900–901, 2005.

Raymond G. Gordon, editor. *Ethnologue: Languages of the World*. SIL International, 15th edition, 2005.

Robbie Haertel, Eric Ringger, Kevin Seppi, James Carroll, and Peter McClanahan. *Assessing the Costs of Sampling Methods in Active Learning for Annotation*. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2008). Columbus, OH, USA, 2008.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. *Learning Bilingual Lexicons from Monolingual Corpora*. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2008). Columbus, OH, USA, 2008.

Dilek Hakkani-Tür, Giuseppe Riccardi, and Gokhan Tur. *An active approach to spoken language processing. ACM Transactions on Speech and Language Processing (TSLP) (October 2006)*, 3(3), 2006.

Almut Silja Hildebrand. Translation Model Adaptation for Statistical Machine Translation using Information Retrieval. Master's thesis, Universität Karlsruhe, 2005.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. *Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval*. In Proceedings of the Conference of the European Association for Machine Translation (EAMT 2005). Budapest, Hungary, 2005.

Roger Hsiao, Ashish Venugopal, Thilo Köhler, Ying Zhang, Paisarn Charoenpornsawat, Andreas Zollmann, Stephan Vogel, Alan W. Black, Tanja Schultz, and Alex Waibel. *Optimizing Components for Handheld Two-way Speech Translation for an English-Iraqi Arabic System*. In Proceedings of the International Conference on Spoken Language Processing (ICSLP 2006). Pittsburgh, PA, USA, 2006.

Fei Huang. *Cluster-specific Name Transliteration*. In Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005). Vancouver, BC, Canada, 2005.

Fei Huang. Multilingual Named Entity Extraction and Translation from Text and Speech. Ph.D. thesis, Carnegie Mellon University, 2006.

Fei Huang, Ying Zhang, and Stephan Vogel. *Mining Key Phrase Translations from Web Corpora*. In Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005). Vancouver, BC, Canada, 2005.

Xuedong Huangxi, Alex Acero, and Hsiao-Wuen Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall PTR, 2001.

David Albert Huffman. *A Method for the Construction of Minimum-Redundancy Codes. Proceedings of the I.R.E.*, pages 1098–1102, 1952.

Rebecca Hwa. *Sample Selection for Statistical Parsing. Computational Linguistics*, 30(3):253–276, 2004.

Ryosuke Isotani, Kiyoshi Yamabana, Shinichi Ando, Ken Hanazawa, Shin-ya Ishikawa, Tadashi Emori, Ken-ichi Iso, Hiroaki Hattori, Akitoshi Okumura, and Takao Watanabe. *An Automatic Speech Translation System on PDAs for Travel Conversation.* In Proceedings of the International Conference on Multimodal Interfaces (ICMI 2002). Pittsburgh, PA, USA, 2002.

Ryosuke Isotani, Kiyoshi Yamabana, Shinichi Ando, Ken Hanazawa, Shin-ya Ishikawa, and Ken-ichi Iso. *Speech-to-speech translation software on PDAs for travel conversation.* Technical report, NEC research & development, Tokyo, Japan, 2003.

J. Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. *Improving Translation Quality by Discarding Most of the Phrasetable.* In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007). Prague, Czech Republic, 2007.

Douglas Jones, Martha Herzog, Hussny Ibrahim, Arvind Jairam, Wade Shen, Edward Gibson, and Michael Emonts. *ILR-Based MT Comprehension Test with Multi-Level Questions.* In Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007). Rochester, NY, USA, 2007.

Teresa M. Kamm and Gerard G. L. Meyer. *Selective sampling of training data for speech recognition.* In Proceedings of the Human Language Technology Conference (HLT 2002), pages 20–24. Morgan Kaufmann Publishers Inc., San Diego, CA, USA, 2002.

Richard M. Karp. *Reducibility Among Combinatorial Problems.* In Complexity of Computer Computations. New York, NY, USA, 1972.

Vipul Kashyap. *The UMLS semantic network and the semantic web.* In Proceedings of the American Medical Informatics Association Symposium (AMIA 2003). Washington, DC, USA, 2003.

Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems.* Springer, 2004.

Ken Kister. *Dictionaries defined. Library Journal*, 117(11):43–46, 1992.

Reinhard Kneser and Hermann Ney. *Improved Backing-Off for M-gram Language Modeling.* In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 1995). Detroit, MI, USA, 1995.

Philipp Koehn. *A Beam Search Decoder for Statistical Machine Translation Models.* In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA 2004). Baltimore, MD, USA, 2004.

Philipp Koehn. *Europarl: A Parallel Corpus for Statistical Machine Translation.* In Proceedings of MT Summit X. Phuket, Thailand, 2005.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. *Moses: Open Source Toolkit for Statistical Machine Translation.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2007). Prague, Czech Republic, 2007.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. *Statistical Phrase-Based Translation.* In Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003). Edmonton, Canada, 2003.

Philippe Langlais and Alexandre Patry. *Translating Unknown Words by Analogical Learning.* In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007). Prague, Czech Republic, 2007.

Alon Lavie, Katharina Probst, Erik Peterson, Stephan Vogel, Lori Levin, Ariadna Font-Llitjos, and Jaime Carbonell. *A Trainable Transfer-based Machine Translation Approach for Languages with Limited Resources.* In Proceedings of the Conference of the European Association for Machine Translation (EAMT 2004). Valletta, Malta, 2004.

Alon Lavie, Stephan Vogel, Lori Levin, Erik Peterson, Katharina Probst, Ariadna Font Llitjos, Rachel Reynolds, Jaime Carbonell, and Richard Cohen. *Experiments with a Hindi-to-English Transfer-based MT System under a Miserly Data Scenario. ACM Transactions on Asian Language Information Processing (TALIP)*, 2(2):143–163, 2003.

Alon Lavie, Alex Waibel, Lori Levin, Michael Finke, Donna Gates, Marsal Gavalda, Torsten Zeppenfeld, and Puming Zhang. *Janus-III: Speech to Speech Translation in Multiple Languages.* In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 1997). Munich, Germany, 1997.

Yves Lepage and Etienne Denoual. *ALEPH: an EBMT system based on the preservation of proportional analogies between sentences across languages.* In Proceedings of the International Workshop for Spoken Language Translation (IWSLT 2005). Pittsburgh, PA, USA, 2005.

C. Lindberg. *The Unified Medical Language System (UMLS) of the National Library of Medicine. Journal of the American Medical Record Association*, 61(5):40–42, 1990.

Yajuan Lü, Jin Huang, and Qun Liu. *Improving Statistical Machine Translation Performance by Training Data Selection and Optimization.* In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007). Prague, Czech Republic, 2007.

Carsten Lund and Mihalis Yannakakis. *On the hardness of approximating minimization problems. Journal of the ACM (JACM)*, 41(5):960–981, 1994.

Coşkun Mermer, Hamza Kaya, and Mehmet Uğur Doğan. *The TÜBITAK-UEKAE Statistical Machine Translation System for IWSLT 2007.* In Proceedings of the International Workshop for Spoken Language Translation (IWSLT 2007). Trento, Italy, 2007.

Lauren Neergard. *Hospitals struggle with growing language barrier. Associated Press, The Charlotte Observer (September 2nd, 2003)*, 2003.

Sonja Niessen, Franz Josef Och, Gregor Leusch, and Hermann Ney. *An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research.* In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2000). Athens, Greece, 2000.

Douglas W. Oard. *The surprise language exercises. ACM Transactions on Asian Language Information Processing (TALIP)*, 2(2):79–84, 2003.

Frans Josef Och, Nicola Ueffing, and Hermann Ney. *An Efficient A* Search Algorithm for Statistical Machine Translation.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2001) - Workshop on Data-Driven Machine Translation. Toulouse, France, 2001.

Franz Josef Och. *Minimum Error Rate in Statistical Machine Translation.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2003). Sapporo, Japan, 2003.

Franz Josef Och and Hermann Ney. *Discriminative training and maximum entropy models for statistical machine translation.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2002). Philadelphia, PA, USA, 2002.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. *BLEU: A method for automatic evaluation of machine translation.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2001). Philadelphia, PA, USA, 2001.

Michael Paul. *Overview of the IWSLT 2006 Evaluation Campaign.* In Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2006). Kyoto, Japan, 2006.

David Pisinger. Algorithms for Knapsack Problems. Ph.D. thesis, University of Copenhagen, 1995.

Katharina Probst and Alon Lavie. *A structurally diverse minimal corpus for eliciting structural mappings between languages.* In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA 2004). Washington, DC, USA, 2004.

Katharina Probst and Lori Levin. *Challenges in automated elicitation of a controlled bilingual corpus.* In Proccedings of the International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2002). Kyoto, Japan, 2002.

Martin Raab. Language Model Techniques in Machine Translation. Master's thesis, Universität Karlsruhe, 2006.

Reinhard Rapp. *Automatic identification of word translations from unrelated English and German corpora.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 1999). College Park, MD, USA, 1999.

Manny Rayner, Pierrette Bouillon, Nikos Chatzichrisafis, Marianne Santaholma, and Marianne Starlander. *MedSLT:A Limited-Domain Unidirectional Grammar-Based Medical Speech Translator.* In Proceedings of First International Workshop on Medical Speech Translation at the Human Language Technology Conference and the Conference of the North American

Chapter of the Association for Computational Linguistics (HLT-NAACL 2006). New York, NY, USA, 2006.

Ran Raz and Muli Safra. *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP.* In Proceedings of the ACM Symposium on the Theory of Computing (STOC 1997). El Paso, TX, USA, 1997.

Jürgen Reichert and Alex Waibel. *The ISL EDTRL System.* In Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2005). Kyoto, Japan, 2004.

Roni Rosenfeld. *Two decades of statistical language modeling: where do we go from here? Proceedings of the IEEE*, 88(8):1270–1278, 2000.

Shirin Saleem, Krishna Subramanian, Rohit Prasad, David Stallard, Chia lin Kao, Prem Natarajan, and Raid Suleiman. *Improvements in Machine Translation for English/Iraqi Speech Translation.* In Proceedings of Interspeech 2007. Antwerp, Belgium, 2007.

Gerard Salton and Christopher Buckley. *Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management*, 24(5):513–523, 1988.

Satoshi Sato and Makoto Nagao. *Toward Memory-based Translation.* In Proceedings of the International Conference on Computational Linguistics (COLING 1990). Helsinki, Finland, 1990.

Craig Schlenoff, Brian Weiss, Michelle Potts Steves, Greg Sanders, Emile Morse, Ann Virts, Tony Downs, John Garofolo, Sebastien Bronsart, Sherri Condon, Jon Phillips, Christy Doran, Dan Parvaz, John Aberdeen, Bea Oshika, Greg Krill, Karine Megerdoomian, Larry Sager, Yale Marc, Marnie Menzel, and Michael Emonts. *Spoken Language Communication and Translation System for Tactical Use (TRANSTAC) - July 2007 Evaluation Report*, 2007. NIST.

Tanja Schultz and Alan Black. *Challenges with Rapid Adaptation of Speech Translation Systems to New Language Pairs.* In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006). Toulouse, France, 2006.

Tanja Schultz and Katrin Kirchhoff, editors. *Multilingual Speech Processing.* Elsevier Academic Press, 2006.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. *Query by Committee*. In Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory (CoLT). Pittsburgh, PA, USA, 1992.

John Simpson and Edmund Weiner, editors. *Oxford English Dictionary*. Clarendon Press, 1989.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. *A Study of Translation Edit Rate with Targeted Human Annotation*. In Proceedings of Association for Machine Translation in the Americas (AMTA 2006). Cambridge, MA, USA, 2006.

Andreas Stolcke. *SRILM - An Extensible Language Modeling Toolkit*. In Proceedings of the International Conference on Spoken Language Processing (ICSLP 2002). Denver, CO, USA, 2002.

Tanaka Takaaki and Yoshihiro Matsuo. *Extraction of translation equivalents from non-parallel corpora*. In Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 1999). Chester, UK, 1999.

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. *Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversations in the Real World*. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2002). Las Palmas, Spain, 2002.

David Talbot and Miles Osborne. *Randomised Language Modelling for Statistical Machine Translation*. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2007). Prague, Czech Republic, 2007a.

David Talbot and Miles Osborne. *Smoothed Bloom filter language models: Tera-Scale LMs on the Cheap*. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007). Prague, Czech Republic, 2007b.

Sebastian Thrun and Knut Moeller. *Active exploration in dynamic environments*. *Advances in Neural Information Processing Systems*, 4:531–538, 1992.

Leslie C. Twilley, Peter Dixon, Dean Taylor, and Karen Clark. *University of Alberta norms of relative meaning frequency for 566 homographs. Memory & Cognition*, 22(1):111–126, 1994.

Nicola Ueffing and Hermann Ney. *Training Corpus Size and Statistical Machine Translation Quality.* In Proceedings of Informatiktage 2002 der Gesellschaft für Informatik. Bad Schussenried, Germany, 2002.

Nicola Ueffing, Franz Josef Och, and Hermann Ney. *Generation of Word Graphs in Statistical Machine Translation.* In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002). Philadelphia, PA, USA, 2002.

Ashish Venugopal, Andreas Zollmann, and Stephan Vogel. *An Efficient Two-Pass Approach to Synchronous-CFG Driven Statistical MT.* In Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007). Rochester, NY, USA, 2007.

Ashish Venugopal, Andreas Zollmann, and Alex Waibel. *Training and Evaluation Error Minimization Rules for Statistical Machine Translation.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2005) - Workshop on Data-driven Machine Translation and Beyond (WPT-05). Ann Arbor, MI, USA, 2005.

Stephan Vogel. *SMT Decoder Dissected: Word Reordering.* In Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE 2003). Beijing, China, 2003.

Stephan Vogel. *PESA: Phrase Pair Extraction as Sentence Splitting.* In Proceedings of MT Summit X. Phuket, Thailand, 2005.

Stephan Vogel and Hermann Ney. *Translation with Cascaded Finite State Transducers.* In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2000). Hongkong, China, 2000.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. *HMM-based Word Alignment in Statistical Translation.* In Proceedings of the International Conference on Computational Linguistics (COLING 1996). Copenhagen, Denmark, 1996.

Clare R. Voss and Calandra R. Tate. *Task-based Evaluation of Machine Translation (MT) Engines: Measuring How Well People Extract Who,*

When, Where-Type Elements in MT Output. In Proceedings of the Conference of the European Association for Machine Translation (EAMT 2005). Oslo, Norway, 2006.

Alex Waibel, Ahmed Badran, Alan W. Black, Robert Frederking, Donna Gates, Alon Lavie, Lori Levin, Kevin Lenzo, Laura Mayfield Tomokiyo, Jürgen Reichert, Tanja Schultz, Dorcas Wallace, Monika Woszczyna, and Jing Zhang. *Speechalator: Two-way Speech-to-Speech Translation in your Hand*. In Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003). Edmonton, Canada, 2003a.

Alex Waibel, Ahmed Badran, Alan W. Black, Robert Frederking, Donna Gates, Alon Lavie, Lori Levin, Kevin Lenzo, Laura Mayfield Tomokiyo, Jürgen Reichert, Tanja Schultz, Dorcas Wallace, Monika Woszczyna, and Jing Zhang. *Speechalator: Two-way Speech-to-Speech Translation on a Consumer PDA*. In Proceedings of Eurospeech 2003. Geneva, Switzerland, 2003b.

Sally Wehmeier, editor. *Oxford Advanced Learner's Dictionary*. Oxford University Press, 7th edition, 2007.

Kiyoshi Yamabana, Seiya Osada, Ken Hanazawa, Akitoshi Okumura, Ryosuke Isotani, and Takao Watanabe. *A speech translation system with mobile wireless clients*. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2003). Sapporo, Japan, 2003.

Kenji Yamada and Kevin Knight. *A Syntax-Based Statistical Translation Model*. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2001). Toulouse, France, 2001.

Luke S. Zettlemoyer and Robert C. Moore. *Selective Phrase Pair Extraction for Improved Statistical Machine Translation*. In Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007). Rochester, NY, USA, 2007.

Rong Zhang and Alexander I. Rudnicky. *A new data selection approach for semi-supervised acoustic modeling*. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006). Toulouse, France, 2006.

Ying Zhang, Fei Huang, and Stephan Vogel. *Mining translations of OOV terms from the web through cross-lingual query expansion.* In Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005). New York, NY, USA, 2005.

Ying Zhang and Stephan Vogel. *Measuring Confidence Intervals for the Machine Translation Evaluation Metrics.* In Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 2004). Baltimore, MD, USA, 2004.

Ying Zhang and Stephan Vogel. *An Efficient Phrase-to-Phrase Alignment Model for Arbitrarily Long Phrase and Large Corpora.* In Proceedings of the Conference of the European Association for Machine Translation (EAMT 2005). The European Association for Machine Translation, Budapest, Hungary, 2005.

Ying Zhang and Stephan Vogel. *Suffix array and its applications in empirical natural language processing.* Technical Report CMU-LTI-06-010, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2006.

Ying Zhang and Stephan Vogel. *PanDoRA: A Large-scale Two-way Statistical Machine Translation System for Hand-held Devices.* In Proceedings of MT Summit XI. Copenhagen, Denmark, 2007.

Bing Zhao, Nguyen Bach, Ian Lane, and Stephan Vogel. *A Log-linear Block Transliteration Model based on Bi-Stream HMMs.* In Proceedings of the Human Language Technology Conference and the Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2007). Rochester, NY, USA, 2007.

Bing Zhao and Alex Waibel. *Learning a LogLinear Model with Bilingual Phrase-Pair Features for Statistical Machine Translation.* In Proceedings of the SigHan Workshop. Jeju Island, Korea, 2005.

Bowen Zhou, Daniel Déchelotte, and Yuqing Gao. *Two-way Speech-to-Speech Translation on Handheld Devices.* In Proceedings of the International Conference on Spoken Language Processing (ICSLP 2004). Jeju Island, Korea, 2004.

Bowen Zhou, Yuqing Gao, Jeffrey Sorensen, Daniel Déchelotte, and Michael Picheny. *A Hand-held Speech to Speech Translation System.* In Proceedings

of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2003). St. Thomas, U.S. Virgin Islands, USA, 2003.

Jacob Ziv and Abraham Lempel. *A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

Matthias Eck

4730 Centre Avenue, Apt. 201
Pittsburgh, PA, 15213
USA

Telefon: ++1-412-478-9187
E-mail: matteck@cs.cmu.edu

Geburtstag: 16. Juli 1977
Geburtsort: Miltenberg

Ausbildung:

Oktober 1997 – Mai 2002
Informatikstudium an der Universität Karlsruhe (TH)
Vertiefungsfächer: Telematik, Graphische Datenverarbeitung
Ergänzungsfach: Operations Research
Diplomarbeit: „Wissensnetze zur Navigation und semantischen Suche in der Internet-basierten Aus- und Weiterbildung“

Abschluss: Diplom-Informatiker, Abschlussnote: 1.2

1987 - 1996
Johannes-Butzbach-Gymnasium Miltenberg

Berufliche Erfahrung:

Seit Januar 2006
Research Programmer
interACT, Carnegie Mellon University, Pittsburgh, USA
- Forschung im Bereich Statistische Machinenübersetzung
- Portabilität auf neue Sprachpaare
- Entwicklung tragbarer Übersetzungssysteme
- Vokabularerweiterung der Übersetzungssysteme
- Übersetzungssysteme für touristische, medizinische und militärische Einsatzgebiete

Juli 2002-Dezember 2005
Wissenschaftlicher Mitarbeiter
interACT, Universität Karlsruhe (TH)
- Forschung im Bereich Statistische Maschinenübersetzung
- Entwicklung von Techniken zur Sprachmodelladaption

August 1999- Mai 2002
Studentische Hilfskraft
Institut für Algorithmen und Kognitive Systeme, Universität Karlsruhe (TH)
- Entwicklung eines Simulators für Quantenschaltkreise

Oktober 1999 - Juli 2001:
Tutor für die Vorlesungen Informatik 1, 2, 3 und 4
Tutor im Praktikum Graphische Datenverarbeitung

November 1998 - Juli 2000:
Tutor beim Studienzentrum für Sehgeschädigte

Computerkenntnisse: Java, C/C++, Perl, PHP, Python, Delphi/Pascal, Visual Basic, HTML, LaTeX, SQL
Windows, Linux

Sprachkenntnisse: Deutsch (Muttersprache)
Englisch (verhandlungssicher)

Stipendium: Mai 2000 – Mai 2002
Stipendiat bei IBM Unternehmensberatung GmbH

Veröffentlichungen:

Matthias Eck, Stephan Vogel, and Alex Waibel. Communicating Unknown Words in Machine Translation, Proceedings of LREC 2008, Marrakech, Morocco, May 2008.

Nguyen Bach, Matthias Eck, Paisarn Charoenpornsawat, Thilo Köhler, Sebastian Stüker, ThuyLinh Nguyen, Roger Hsiao, Alex Waibel, Stephan Vogel, Tanja Schultz, Alan W. Black. The CMU TransTac 2007 Eyes-free and Hands-free Two-way Speech-to-Speech Translation System, Proceedings of IWSLT 2007, Trento, Italy, October 2007.

Matthias Eck, Stephan Vogel, and Alex Waibel. Estimating Phrase Pair Relevance for Translation Model Pruning, Proceedings of MT Summit XI, Copenhagen, Denmark, September 2007.

Matthias Eck, Stephan Vogel, and Alex Waibel. Translation Model Pruning via Usage Statistics for Statistical Machine Translation, Proceedings of HLT 2007, Rochester, NY, April 2007.

Matthias Eck, Ian Lane, Nguyen Bach, Sanjika Hewavitharana, Muntsin Kolss, Bing Zhao, Almut Silja Hildebrand, Stephan Vogel, and Alex Waibel. The UKA/CMU Statistical Machine Translation System for IWSLT 2006, Proceedings of IWSLT 2006, Kyoto, Japan, November 2006.

Matthias Eck, Stephan Vogel, and Alex Waibel. A Flexible Online Server for Machine Translation Evaluation, Proceedings of EAMT 2006, Oslo, Norway, June 2006.

Matthias Eck and Chiori Hori. Overview of the IWSLT 2005 Evaluation Campaign, Proceedings of IWSLT 2005, Pittsburgh, USA, October 2005.

Matthias Eck, Stephan Vogel, and Alex Waibel. Low Cost Portability for Statistical Machine Translation based on N-gram Frequency and TF-IDF, Proceedings of IWSLT 2005, Pittsburgh, USA, October 2005.

Matthias Eck, Stephan Vogel, and Alex Waibel. Low Cost Portability for Statistical Machine Translation based on N-gram Coverage, Proceedings of MT Summit X, Phuket, Thailand, September 2005.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval, Proceedings of EAMT 2005, Budapest, Hungary, May 2005.

Matthias Eck, Stephan Vogel, and Alex Waibel. Improving Statistical Machine Translation in the Medical Domain using the Unified Medical Language System, Proceedings of Coling 2004, Geneva, Switzerland, August 2004.

Bing Zhao, Matthias Eck, and Stephan Vogel. Language Model Adaptation for Statistical Machine Translation via Structured Query Models, Proceedings of Coling 2004, Geneva, Switzerland, August 2004.

Matthias Eck, Stephan Vogel, and Alex Waibel. Language Model Adaptation for Statistical Machine Translation based on Information Retrieval, Proceedings of LREC 2004, Lisbon, Portugal, May 2004.

Karsten Krutz, Matthias Eck, Christian Mayerl, Matthias Riechmann, and Sebastian Abeck. Semantische Suche zur Unterstützung des Internet-basierten Wissenstransfers, e-Learning Workshop KIVS, Leipzig, Germany, 2003.