HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Marcus Dobrinkat

Domain Adaptation in Statistical Machine Translation Systems via User Feedback

Espoo, November 25, 2008

Supervisor: Doc. Timo Honkela, Ph.D., Chief Research Scientist
Instructor: Jaakko Väyrynen, M.Sc. (Tech.)

HELSINKI UNIVERSITY OF TECHNOLOGY
Faculty of Information and Natural Sciences
Degree Programme in Computer Science and Engineering

ABSTRACT OF MASTER'S THESIS

Author: Marcus Dobrinkat

Title of thesis: Domain Adaptation in Statistical Machine Translation Systems via User Feedback

Date: November 25, 2008 **Pages:** 12 + 91

Professorship: T-61

Supervisor: Doc. Timo Honkela, Ph.D., Chief Research Scientist
Instructor: Jaakko Väyrynen, M.Sc. (Tech.)

Machine translation research has progressed in recent years thanks to statistical machine learning methods, sufficient computational power, open source tools and increasing availability of bilingual parallel text resources. However, most of these systems stay in the hands of researchers and are not improved with public users in mind. The motivation behind this thesis is the vision of freely available machine translation systems. They may be particularly important for languages and domains where there is not enough commercial interest for providing such services otherwise.

The main focus of this work was to collect reference translations for Finnish news sentences, and to use this data to improve a baseline translation system on this news domain. A web application was created for rating and correcting translations and volunteers were invited to participate the effort. Then, three different approaches to domain adaptation were realized and evaluated using the news domain data. In particular, language and translation model interpolation and post-editing have been studied. Thanks to volunteers, a 1 000 sentence bilingual Finnish-English news corpus was assembled. The corpus is a good asset for further research in domain adaptation. The adaptation results show that a combination of language model and translation model interpolation effectively adapts the baseline system to the news domain. Using available domain adaptation methods, translation systems can be built with simple means and adjusted to the users' needs by community feedback.

Keywords: Statistical Machine Translation, Domain Adaptation, Evaluation

Acknowledgments

I have written this thesis in the computational cognitive systems group, which I wish to thank for the warm welcome and the good, inspiring working atmosphere. I am especially thankful to Jaakko Väyrynen, who has been a devoted and obliging instructor. In our discussions and through your countless feedback, I have learned a lot and you helped me to focus. Next, I would like to thank my supervisor, Doc. Timo Honkela for his innovative ideas, feedback and help with searching a suitable topic.

I am grateful to my colleagues Sami Virpioja, for helping me with Morfessor and to Janne Argillander for proofreading and interesting discussions.

I also wish to thank IBM Finland for the financial support and work time flexibility, which permitted me to write this thesis.

My family has provided me with constant help and love. I am especially grateful to my wife Sonja, who managed to support me during this long one year. Thanks to Ole and Synne, who spent countless hours taking care of their grandchildren, and to my parents Gitti and Gert-Ulrich, who always encouraged me.

Then, a big thank you to all the volunteers, who translated news sentences from Finnish to English to collect a bilingual corpus: jaakkov, jargilla, laban, timo, marisa, mpolla, sonja, jmertane, svirpioj, okohonen, tino, Ole, Jum, jedilover, jatoivol, Tim, tlindh, jhimberg, thirsima and lindarella.

Espoo, November 25, 2008

Marcus Dobrinkat

Abbreviations and Acronyms

APE	Automatic Post Editing
BLEU	Bilingual Language Evaluation Understudy
BP	Brevity Penalty
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
CGI	Common Gateway Interface
CPU	Central Processing Unit
Europarl	A Parallel Corpus for Statistical Machine Translation
GTM	General Text Matcher
IT	Information Technology
LM	Language Model
MAP	Maximum A Posteriori
MB	Megabyte
MDL	Minimum Description Length Principle
METEOR	Metric for Evaluation of Translation with Explicit ORdering
MT	Machine Translation
NIST	National Institute of Standards and Technology
NON	No-morph Category
PBSMT	Phrase Based Statistical Machine Translation
PER	Position Independent Error Rate
PRE	Prefix
PSVC	Problem Solving Virtual Community
RAM	Random Access Memory
RBMT	Rule Based Machine Translation System
RM	Reordering Model
RSD	Relative Standard Deviation
SL	Source Language
SMT	Statistical Machine Translation
SPE	Statistical Post Editing
STM	Stem
SUF	Suffix
TL	Target Language
TM	Translation Model
UI	User Interface
WER	Word Error Rate

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine translation (MT) is the translation of text of one natural language into another natural language with the help of a computer program. The pioneers of MT research started as early as 1933 and the interest in this technology had a boom in the 1950s. A lack in prospect of machine translation in 1966 (Automatic Language Processing Advisory Committee, 1966) caused fund withdrawal and stagnation of research.

Limited domain MT systems were successful in the 1960s. Later research and various commercial MT systems started to flourish but the big breakthrough has yet to come.

For some popular language pairs such as English-French or German-Russian, rule based machine translation systems (RBMT) have reached a good quality. limited research, rule based systems are less available (Hutchins, 1995). For these languages, statistical machine translation (SMT) provides an alternative path, which has, however, not yet delivered satisfactory results.

SMT systems utilize large bilingual corpora, which are texts from one language paired with their translations in another language. Each sentence pair helps to build up a dictionary of word-to-word translation probabilities. A word-to-word translation pair is assigned a higher likelihood if it appears in several sentence pairs.

A dictionary of probabilities for word-to-word translations is the result of this iterative process. These are then used to heuristically form a multiple word (phrase) dictionary. For achieving translation quality that is good enough for manual post-editing, a very large corpus far exceeding the about 1 million sentence pairs that we have used in our research, is required. The quality of the translations rises with the corpus size.

Corpus based research nowadays has increased thanks to the huge amount of text, that the Internet makes available for many languages. Large bilingual corpora arise from company reports and government proceedings in multilingual countries like Canada, Switzerland, Finland, Belgium, Singapore or communities like the European Union. (Koehn, 2005).

Given the required bilingual resources, SMT systems are faster and cheaper to develop than RBMT systems, as they avoid the immense work of linguists creating language dependent processing modules (parsers, taggers, etc.), required for syntactic and semantic transfer.

The effort invested in MT research and the relatively modest results already show, that MT is one of the more difficult problems in computational linguistics. For successful MT, several natural language processing problems have to be solved: named entity recognition, part-of-speech tagging, morphological analysis and generation, disambiguation and structural transfer just to name a few that are closely related to logical or rule-based approaches.

When concentrating on statistical approaches, the focus lies on a sound statistical model that can efficiently be used with large corpora. The other important ingredient is a large bilingual corpus of good quality.

More and more freely available translation systems exist today providing acceptable quality. However, in languages and domains with small target audience, there is little incentive for investment as that market segment is very small and lacks growth potential.

This thesis provides a start towards my vision to create a freely available translation platform, which can be improved by the user community by adding and correcting translations. My motivation comes from the spirit of open source, access to free information and in particular from the fact that no free translation system exists for Finnish providing acceptable translation quality.

As explained earlier, the problem of MT is no trivial one. One source of inspiration for breaking the problem down into smaller parts was Luis von Ahn's work (von Ahn, 2006). With his software, he motivates people to use their brain power for solving open problems in artificial intelligence, may it be tagging images (von Ahn and Dabbish, 2004), recognizing words (von Ahn et al., 2003) or locating objects in images (von Ahn et al., 2006): tasks that are all hard to solve for computer programs.
The basic concept is that volunteer users create additional training samples that help to solve a difficult problem. Machine learning programs can then generalize the collected knowledge and finally automate the task. The mo-

tivation for users comes from the fascinating appeal of a multi-user game such as Peekaboom (von Ahn et al., 2006) or it can be something very useful instead, such as protecting web pages from spam using a "human only test" (von Ahn et al., 2003). The latter is often used, because it is provided as a free service by Carnegie Mellon University for improving web security.

Using these ideas in the context of MT would allow users to customize their MT system. This would also alleviate the problem that MT users often do not like the output of MT systems, because they say the quality were not good enough. One of the reasons is the difference in domain: often the type of language that was used to train the system is different from the one humans expect, and use in everyday life.

This work focuses on collecting news domain data, domain adaptation of statistical machine translation, evaluation of MT quality, automatic post-editing and touches communities and human computation.

1.1 Thesis Objectives

This thesis seeks to advance the work towards the vision of improving machine translation quality by

1. Creation of the tools and data required to test new domain adaptation approaches, which includes a test corpus for Iltalehti news domain data.
2. Answer to the question of how user feedback effectively serves domain adaptation.
3. Confirmation that human and automatic scores such as BLEU correlate.
4. Examination of what is required to motivate people to voluntarily participate in a community effort.

1.2 Structure of the Thesis

In the second Chapter of this thesis, I lay out the theoretical methodology required for the experiments. A description of the domain adaptation experiments follows in Chapter 3. Chapter 4 reports the results, which are discussed in Chapter 5. The final Chapter 6 concludes the thesis.

Chapter 2

Theoretical Background

The theoretical background of this work is presented in three parts. Section 2.1, methodology, introduces important concepts that are related to the core tasks of this work. The following two topics are central and therefore parts on their own. Section 2.2, MT Evaluation introduces methods to assess MT output. The last part, domain adaptation, surveys existing approaches to improve MT performance for a shifted domain (Section 2.3).

2.1 Methodology

This section contains a number of important background concepts. Section 2.1.1 discusses, what is needed to make people participate voluntarily in community projects. Then, Section 2.1.2 introduces phrase based statistical machine translation, which is the basis for the MT system used in our experiments. In the first practical part of this thesis, volunteers evaluate morph based translations. The basics of unsupervised morph segmentation are explained in Section 2.1.3. For domain adaptation, text corpora from different domains are used. Section 2.1.4 discusses, what a domain means, and how differences between text corpora can be measured.

2.1.1 Employing Volunteers for Problem Solving

How can people be motivated to voluntarily participate in some project? We approach the problem by first looking at the motives of virtual community participants. In virtual communities, people consciously spend their time to advance the community towards its advertised goal.

Then we look at the more specific topic of "Games with a purpose". Here, the real problem is covered (although not secretly) by a more stimulating task in the form of a game. The real problem is often laborious and little exciting; therefore working with it directly would not attract people.

Virtual Communities

A virtual community is a group of people that communicate and exchange information via a communication medium, typically the Internet (on-line community) in form of emails, Usenet, forums, wikis or some other web based collaboration application instead of personal meetings.

With the rise of on-line communities from Usenet to Wikipedia and also the commercial success of open source software, researchers have investigated the motivating factors behind the voluntary participation with seemingly no monetary gain.

Bitzer et al. (2007) use a private provision of public goods model to find agents with a high motivation to work on open source software. Empirical research supports their model; the authors show that the main properties of such highly motivated programmers are:

- High gain from using the software — the person has a need for the specific software.
- Attraction to higher reputation — the persons wishes to be accepted as a part of the community and gain a higher status.
- High gratification from play — the fun to master a problem or just to play with it.
- Attraction to higher wage
- Young individual
- High programming skills

Another work from the knowledge management domain provides a more general view on the motivating factors in on-line communities (which they call problem solving virtual community — PSVC). Yu et al. (2007) propose a motivational model for PSVC contribution. Individual motivations act together to form a knowledge contribution intention which consists of:

Self interest motivation:

- enhancement
- active learning
- reputation
- enjoyment of helping others
- self-protection
- reciprocity

public interest motivation:

- moral obligation
- advancement of virtual community

Although there are many success stories of open source projects and on-line communities, there are even more failed projects of that kind. The reasons behind such failures have been researched as well as success factors, which include off-line interactions among members, the supply of content that viewers perceive as useful and a decent IT Infrastructure (Koh et al., 2007).

Games with a Purpose

The goal of "Games with a purpose" is to solve problems that require vast amounts of intelligent human input. This kind of data is expensive to gather when one needs to employ humans for the task. "Games with a purpose" hide the real task behind games or other applications that attract people or have some other added value for them. Examples are the Google Image Labeler and CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart). CAPTCHAs are distorted characters that are placed on web sites to improve security by guaranteeing human user access, which prevents spam. The characters are distorted such that they are hard to recognize by programs, but still easy to recognize for humans. Writing the clear text then gives access to the web site services. The hidden use is that annotated data is collected (the image plus the clear text), which helps to improve optical character recognition.

Googles Image Labeler works with a different motivation. Here, fun is the main aspect. A user is paired with another remote user and presented an image. The users then write words that describe what they see on the image. When both users enter the same words, their score increases. The hidden goal is to improve image search by collecting more human annotated images, which is a very expensive task otherwise.

2.1.2 Phrase Based Statistical Machine Translation

When translating from a source language (SL) to a target language (TL), a source sentence $s = w_1, ..., w_j, ..., w_J$ is rendered as target sentence $t = w_1, ..., w_i, ..., w_I$. In statistical machine translation (SMT), a statistical model governs the mapping of source to target sentence. Although original ideas of SMT can be found as early as 1949 (Weaver, 1949), SMT first rose with the influential work of Brown et al. (1994). Brown et al. presented the source-channel approach for machine translation (Equation 2.1).

$$P(t|s) = \frac{P(s|t)\ P(t)}{P(s)} \tag{2.1}$$

$$\hat{t} = \arg\max_{t}\ P(s|t)\ P(t) \tag{2.2}$$

As such, the problem is split into two simpler sub-problems. The best translation is found with a global search (the arg-max), using the translation model $P(s|t)$ and the language model $P(t)$. As the exact probability is not required, the best translation can be found by maximizing Equation 2.1, which allows us to drop the denominator (Equation 2.2).

This model does not allow straightforward inclusion of additional features. A more general approach to find $P(t|s)$ has been formulated by Och and Ney (2001). They use a log-linear model, employing a maximum entropy framework, which provides M feature functions $h_m(t, s)$ and a weight λ_m for each feature. The translation probability $P(t|s)$ is then defined as:

$$P(t|s) = \frac{\exp\left[\sum_{m=1}^{M}\ \lambda_m\ h_m(t,s)\right]}{\sum_{t'}\ \exp\left[\sum_{m=1}^{M}\ \lambda_m\ h_m(t',s)\right]} \tag{2.3}$$

As before, the best translation is found when $P(t|s)$ is maximized:

$$\hat{t} = \arg\max_{t} \; P(t|s) \tag{2.4}$$

When substituting 2.3 into 2.4, the renormalization in the denominator can be dropped:

$$\hat{t} = \arg\max_{t} \; \sum_{m=1}^{M} \lambda_m \; h_m(t, s) \tag{2.5}$$

The source-channel approach of Brown et al. can be modeled as a special case of this one, by choosing the translation model and the language model as two features with equal weight ($\lambda_1 = \lambda_2 = 1$):

$$h_1(t, s) = \log \; p_{\hat{\gamma}}(t) \tag{2.6}$$

$$h_2(t, s) = \log \; p_{\hat{\theta}}(s|t) \tag{2.7}$$

More details about how the model is discriminatively trained, can be found in Och and Ney (2001). One advantage of this more general model is that additional features can easily be added. Some typical features (Koehn et al., 2003; Och and Ney, 2001) are:

- A phrase translation $phi(t|s)$ and the reverse $phi(s|t)$, which tell in both directions how likely two phrases translate to each other.
- A language model $LM(t)$, which tells if the candidate translation is a proper sentence of the target language.
- A word penalty $W(t)$, which tries to avoid too long or too short candidate sentences.
- A reordering (or distortion) model $D(s, t)$, which allows reordering of the phrases and favors translation candidates with the proper phrase order for the target language.
- A lexical weight $p_w(s|t, a)$, which tells how well the single words in a candidate phrase alignment a translate to each other. This model can be created from the extracted word alignment or a commercial dictionary.

Phrase Translation Model Using only a word dictionary for translation, results in problems when the number of words are different in source and target language. This is caused by differences in the languages regarding compound words, morphology or idioms. Storing consecutive words that belong together (called a phrase) has shown to alleviate these effects. Therefore, instead of storing words, phrases take their place in the dictionary (the phrase table). As shown by Koehn et al. (2003), using a phrase length of up to 3 words is sufficient. In the approach used by Koehn et al. (2003), the phrase dictionary is extracted from two-way word dictionaries as basis (obtained by statistical word alignment). A set of heuristics ("rules of thumb") is applied to grow phrases. In contrast, a mathematically proven approach, such as the joint-probability model for phrase translation (Marcu and Wong, 2002), might lead to a better solution, but is computationally much more expensive. Other approaches to phrase translation are shown by Och et al. (1999) and Yamada and Knight (2001).

Language Model A language model is used to predict the next word or tell how likely a sequence of words $w_1, ..., w_n$ is. It is typically created from word co-occurence statistics in a large text corpus that represents the target language. In order to avoid overfitting, the length of the word sequence is usually restricted to small n. An n^{th} order language model is called n-gram language model. Due to data sparseness, smoothing techniques are usually applied that improve performance on unseen word sequences. The language model is an important factor in the machine translation model to ensure the intelligibility of the target sentence.

Related Problems

Sentence Alignment Sentence alignment is the task to find sentences in one language that correspond to sentences in the other language. A bilingual corpus serves as input, which has to be already split into separate sentences. Sentence alignment is an extensively studied problem; an overview including a number of different methods can be found in Manning and Schütze (1999) or Mikheev (2003).

One example method is the sentence alignment method by Gale and Church (1993). It is one of the earlier methods, which is relatively simple and works well for close languages and literal translations. The basic assumption of length-based methods is that the translation of some source sentence results in a target sentence with similar total character length. The best alignment

is found by the use of a dynamic programming algorithm, which is a way to plan a multi-stage process in an optimal manner. It is applicable when the problem can be split into similar sub problems and works by first solving the smallest sub problems directly and storing intermediate results that are looked up when needed again to avoid expensive recursion.

Word Alignment Having a bilingual pair of corresponding sentences, word alignment methods are used to find which words correspond to each other. The analysis of which words tend to be translated by which other words, results in a bilingual dictionary. One popular set of methods for word alignment is contained in the IBM model (Brown et al., 1994). A comparison of different methods for statistical word alignment can be found in (Och and Ney, 2003).

2.1.3 Automatic Morphological Segmentation

Morphological segmentation is the process of splitting words into smaller units, which are called morphs. Morphs are realizations of morphemes, which are the smallest meaning-bearing units in a natural language. For example, the word `undertake` can be split into `under + take` each of which is a morph. `take` as well as `took` are realizations of the morpheme `take`. Automatic morphological segmentation means that this analysis is accomplished without human intervention, the only requirement being a monolingual text corpus with examples of natural text for the purpose of training the method.

Morfessor

Morfessor is a tool for unsupervised segmentation of words into morphs (Creutz and Lagus, 2007). There are several models; the simpler baseline model is also used in the improved and more complex categories MAP model as initial step. Using a reference corpus, Morfessor trains segmentation models that contain information about morph appearance (form) and occurrence frequency (usage).

Baseline Model The Baseline model uses the minimum description length (MDL) principle. At start, the algorithm considers all word types as morphs, then iteratively refines the morph lexicon by finding morphs that are a part of other morphs and segmenting the latter further. For example, given the morph list {consistently, recent, recently} and analyzing the word “recently”,

we can add the new morph “ly" and expand the morph list to become {consistently, ly, recent, recently}. Further on, analyzing “consistently”, we add “consistent”, so the list becomes {consistent, consistently, ly, recent, recently}.

Categories MAP Model The Categories MAP model uses maximum a posteriori as a maximization criterion. Morphs are represented as trees, where the form of a morph is either a sequence of letters or two sub-morphs.

Morphs are assigned one of four categories: prefixes (PRE), stems (STM) and suffixes (SUF). One additional temporary category is the no-morph (NON) category. Category transitions are limited such that suffixes cannot appear at the beginning of a word, prefixes cannot appear at the end of a word and a suffix cannot directly follow a prefix. Figure 2.1 shows how the Categories MAP model represents the segmentation of the word "straightforwardness". More details about the Categories MAP Model can be found in Creutz and Lagus (2007).

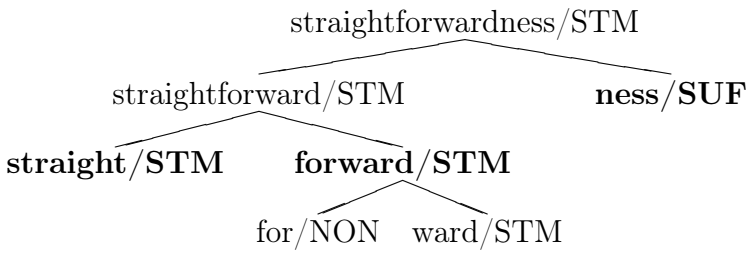


Figure 2.1: Example of how the Categories MAP model represents the segmentation of the word straightforwardness. Every morph has its category assigned as either prefix (PRE), stem (STM), suffix (SUF) or no-morph (NON). The best segmentation of the word, here printed in bold, is the one that does not contain NON-categories.

2.1.4 Comparing Corpora

Statistical machine translation systems are trained with large bilingual text corpora. During evaluation, a test corpus is used that is distinct from the training corpus. Often the test corpus is one part of the original corpus and therefore belongs to the same domain. If we want to evaluate SMT quality in a different domain than the one for which the system was trained, an in-domain corpus for that particular domain is required. Out-of-domain describes the area of knowledge, which is not subject of the current test

set. Measuring the domain of a text it no trivial task. There are many different characteristics in which two texts can differ. For example spoken language versus written language or formal versus informal style. Moneglia (2004) lists some possible domains: news, talk shows, scientific press, teaching, preaching, sport, private conversations, interviews, reportage, business, law, political debate and political speech. The goal of comparing corpora is to find out what two corpora separate or share with respect to certain features. The result of that comparison could be expressed as a measure of similarity or a set of mutual and distinct domains.

Various measures exist that can be used to test for differences between texts. Rayson (2003) and Kilgarriff (2001) review a number of them including χ^2-test, Mann-Whitney ranks test, Student's t-test, log-likelihood ratio test and mutual information statistics.

At least in principle, some difficulties exist with using statistical hypothesis tests for word frequency counts. Usually the null hypothesis states that two phenomena are independent (here the word frequency counts in two different corpora). Language is created with a purpose in mind, therefore it is not random and it is just a matter of having enough data available to be able to reject the null hypothesis (Kilgarriff, 2005; Evert, 2006). This circumstance is aggravated for analyzing linguistics when using large corpora and due to the fact that a large proportion of words in a corpus occur with a low frequency (Zipf's law).

Despite these issues, statistical tests are widely applied and have proven to be useful. Such tests highlight the cases with the strongest evidence of dependency. They are used to support the final goal of learning something about linguistic phenomena rather than estimating exact statistical parameters (Evert, 2006).

Log-likelihood Ratio Hypothesis Test

One possible way to compare two texts is to use the log-likelihood ratio test statistics (Dunning, 1993), which gives more accurate estimates for sparse data (low word counts) than the simpler χ^2-test. A common method for performing statistical tests on text data is to investigate the connection between two random variables using a contingency table. For each word type, an own contingency table holds the observed word frequency counts of both corpora (see Table 2.1). The contingency table contains word type occurrence counts for word type w for each of the corpora in the first row. The second row holds the counts of another word type than w appearing in the corpora. On

the margins are the row and column sums.

Table 2.1: Contingency table for observed word type counts of word type w for two corpora to be evaluated.

	$Corpus_1$	$Corpus_2$	TOTAL
word w	O_{11}	O_{12}	R_1
not word w	O_{21}	O_{22}	R_2
TOTAL	C_1	C_2	N

A table similar to the one for the observed values is created for the expected values as well. These calculate the relative frequencies as if there was one big corpus and then scale them to the size of $Corpus_1$ or $Corpus_2$ as shown in Equation 2.8.

$$E_{11} = C_1\frac{R_1}{N} \qquad E_{12} = C_2\frac{R_1}{N} \qquad E_{21} = C_1\frac{R_2}{N} \qquad E_{22} = C_2\frac{R_2}{N} \tag{2.8}$$

We then formulate the null hypothesis H_0, which states that the observed values O_{ij} result from a random sample of the population, defined by the expected values E_{ij}. This means that the occurrence of the word type is independent of the corpus (Equation 2.9). The alternative hypothesis instead states that the word type is dependent on the corpus, therefore the observed values are a result from a random sample of a different population than the expected values (Equation 2.10).

$$H_0 : \ P(w|Corpus_1) = p = P(w|Corpus_2) \tag{2.9}$$

$$H_1 : \ P(w|Corpus_1) = p_1 \neq p_2 = P(w|Corpus_2) \tag{2.10}$$

In likelihood ratio testing, the likelihood of the two hypotheses is compared by calculating the ratio between them.

$$\lambda = \frac{L(H_0)}{L(H_1)} \tag{2.11}$$

The result value of $G^2 = -2log\lambda$ (ll-value) is asymptotically approaching a χ^2 distribution and can therefore be compared to the χ^2 statistics. Due to

this, the critical value for a certain significance level can be obtained from a χ^2 distribution table using one degree of freedom for our 2x2 contingency table. H_0 will then be rejected for ll-values larger than that critical value.

Assuming a binomial distribution for the hypotheses and evaluating G^2 by using maximum likelihood parameters results in Equation 2.12:

$$G^2 = 2 \sum_{ij} O_{ij} \log \frac{O_{ij}}{E_{ij}} \tag{2.12}$$

Typically, the word types are ranked by decreasing ll-value which exposes the most significant words on the top of the list. As the log-likelihood test is a two-sided test, the ll-value is always positive and does not tell for which corpus the word is characteristic. However that can easily be seen from the words relative frequency.

2.2 Machine Translation Evaluation

Evaluation of machine translation is the process of careful study and judgment of machine translated text regarding its quality, character or degree of excellence. What is the intention behind MT evaluation?

- Allows the comparison of different MT systems or different versions of one system. Evaluation helps to determine which system is the best in a certain aspect or for some specific purpose or domain.
- Allows optimization of performance by finding system modifications that yield improved evaluation results.

The goodness of a translation can be determined by the degree to which the amount of accurate meaning of the original is reproduced (Miller and Beebe-Center, 1956). Gerber (2001) puts it in a similar way: in order to measure the quality of machine translation we should be able to measure the content of text. He argues that text characterization should include the "real-world state of affairs" as well as the "communicative goal" of a piece of text.

Almost 50 years ago it was correct that computers could not understand text, as Bar-Hillel (1960) argues. Artificial intelligence has come a long way so that this statement is gradually disproven. An area closely related to language understanding is question answering. Looking at the 2006 results of the annual text retrieval conference for the question-answering track, shows that

the best systems yield just under 60% correct answers (Dang et al., 2007). This result shows that there is still a long way to go and understanding text is still a hard problem. Natural language structure is very ambiguous and the same holds for word senses, and stylistic properties, which are both hard to capture.

Even though computers cannot fully understand text, we can still make them extract certain text features and let them compare source with target text and target text with reference translations. Following these paradigms, a number of methods have proven useful and work reasonably well in practice. We will explore some of them in Section 2.2.3 when talking about automatic evaluation methods.

2.2.1 Aspects of Evaluation

Miller and Beebe-Center (1956) has studied some psychological methods for MT evaluation. Quality scales for the evaluation of translations should be equally valid for all translations, whether made by humans or by computer systems.

Evaluation criteria are grouped into macro and micro evaluation by van Slype (1979). Macro evaluation considers evaluation aspects with regard to the user requirements: the aspects assess the goodness of a translation, whereas micro evaluation consider the sources of insufficiency and so tries to look inside the translation system black box. In the following, we concentrate on macro evaluation and its different aspects.

Evaluation on the cognitive level measures effectiveness of information and knowledge transfer.

Intelligibility measures the ease with which a translation can be understood. Alternatives are comprehensibility, readability and clarity.

Fidelity measures how much of the information in the source language is successfully transferred to the target language.

Coherence uses a larger example of translated text and measures if the text has a clear and logical structure and is understandable. With an adequate amount of text, a totally wrong translation would not very likely give a coherent text. This does not require the original and can be performed by a monolingual evaluator.

Usability measures how well the translation fits the domain or aspect that the user requires. For example, a translator might see a translation as adequate, as basis to be corrected or as useless.

Evaluation on the economic level measures time-efficiency.

Reading time can be defined as the time required for reading the translated text. Another definition is the ratio of the time required for reading the translation to the time required for reading the original.

Correction time measures the post-editing difficulty by taking the time required for correcting the translated text.

Translation time measures the time required for translation from source to target text.

Evaluation on the linguistic level measures conformity with a linguistic model.

Semantic relationships measures how many semantic relationships are correctly and incorrectly reconstructed by the translation.

Lexical evaluation measures the amount of common words between the reference translation and the translation to be evaluated.

Syntactic and morphological coherence measures how consistent syntax and morphology are.

2.2.2 Human Evaluation

Human evaluation is nowadays mainly done as meta-evaluation for automatic machine translation performance evaluation methods. Human evaluation is labor intensive and time consuming and therefore expensive. Ratings of humans are subjective, as evaluators have different preferences, experience and world knowledge and thus rate differently.

The ALPAC report (Automatic Language Processing Advisory Committee, 1966) suggested intelligibility and fidelity to be the fundamental dimensions for assessing translation adequacy. Based on this finding, Carroll (1966) created a rating scale. They decided to measure fidelity as informativeness of the original. Having read the translation, judges assess if the source sentence

contains any additional knowledge. The 9-point rating scales were created using a laborious, but sound methodology. Several hundred sentences were grouped into 9 heaps of improving intelligibility and fidelity, using the equal-appearing intervals technique. Then, appropriate descriptions for each group were found and iteratively refined. Given that the selection of raters and the rating procedure was conducted very accurately as well, this example shows that much time and human resources can be spent to properly carry out human evaluation.

Other possibilities for human evaluation are to indirectly measure the information content of the translation, using a knowledge test. Readability has been measured by the Cloze test (van Slype, 1979), for which words in a translation are removed and judges subsequently have to fill the blanks. One method to determine which translation of several is better, is to let judges do a pairwise comparison and rank the translations.

As one human evaluation scale, which is commonly used in MT competitions, we present the NIST evaluation approach in the following.

NIST Evaluation Scale This 5-point scale for fluency and adequacy was developed by the Linguistics Data Consortium for the use in the NIST Machine Translation Evaluation Workshop (LDC, 2005). Fluency is similar to intelligibility and adequacy is similar to accuracy or fidelity. The reviewer first rates fluency, then adequacy, by answering the questions in Table 2.2:

Table 2.2: NIST human evaluation rating scale for fluency and adequacy.

Question	Rating Scale
How do you judge the fluency of this translation?	5 – Flawless English 4 – Good English 3 – Non-native English 2 – Disfluent English 1 – Incomprehensible
How much of the meaning expressed in the gold-standard translation is also expressed in the target translation?	5 – All 4 – Most 3 – Much 2 – Little 1 – None

Judges must be native English speakers and have university level education. Before the task, a short training using the assessment software must be performed. The rated segments are presented in order and are extracted from a continuous story.

2.2.3 Automatic Evaluation

Human evaluation is extensive: it assesses many aspects of translation, among others adequacy, intelligibility and accuracy. But it is also expensive and time-consuming. This is a dilemma for developers of machine translation systems. They want to try out, which new ideas improve translation quality on a daily basis. This would be impossible if they had to wait for weeks to obtain human assessments of modified translations.

The goal of automatic evaluation is to create a measure that would mimic human evaluation as closely as possible and that is easy and fast to compute. Another desired property is language independence.

All automatic methods have in common, that they need one or several reference translations. As translation is an open task, there always exist multiple correct solutions. The quality of a reference set always influences the judgments. Therefore a good reference set is one that also includes translation variability, which means multiple corrections.

Existing methods can be grouped into edit distance methods, precision oriented methods, recall oriented methods and methods that harmonize precision and recall. In the following, BLEU is presented as most commonly used automatic scoring method. Then, a summary of other existing methods follows.

Bilingual Evaluation Understudy (BLEU)

The BLEU (Bilingual Language Evaluation Understudy) score was developed in the IBM labs (Papineni et al., 2001) to obtain a rapid and economical way to automatically evaluate machine translation. The score is designed to highly correlate with human assessment. For this work some experiments were conducted for sentence level evaluation using the BLEU score. We decided, however, not to use these experiments further on for evaluation.

The basic idea of BLEU is to reward closeness to one of the human reference translations, using modified unigram precision. The precision is determined by the weighted overlap of n-grams from the candidate translation to the reference translations (for $n = 1, .., 4$). The final score between 0 and 1 tells how close the candidate is to any of the references. BLEU is currently the most commonly used score for comparing MT systems and evaluating improvements, because it is easy to compute and provides reasonable performance. Often, BLEU scores are given as BLEU% from 0 to 100 and are sometimes called BLEU points.

In modified n-gram precision, the numerator is bound to the maximum number of occurrences of that n-gram in any of the references. In other words: of one specific n-gram, take the amount of as many occurrences in the candidate count as can be found in the best matching reference. Let this be the clip count of the n-gram $w_1 \cdots w_n$, defined as:

$$\text{clip\_count}(w_1 \cdots w_n) = \min [\, \text{count}_C(w_1 \cdots w_n) \,,\, \text{count}_{R_m}(w_1 \cdots w_n) \,]$$

With m being the index to the reference translation with the maximum n-gram count:

$$m = \underset{i}{\text{argmax}} [\, \text{count}_{R_i}(w_1 \cdots w_n) \,]$$

The clip count divided by the number of total n-grams in the candidate gives the modified n-gram precision p_n for one single sentence:

$$p_n = \frac{\text{clip\_count}_C(w_1 \cdots w_n)}{\text{count}_C(w_1 \cdots w_n)}$$

This way, longer sentences that repeat correct words are handled nicely. As this property does not effect shorter sentences, the BLEU score uses an explicit punishment for short sentences, the brevity penalty BP:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{1-\frac{r}{c}} & \text{if } c \leq r \end{cases}$$

Typically, recall is used to prevent the mentioned short translations by using the n-gram count of the reference as denominator. However when having multiple reference translations it is not trivial to formulate recall, therefore the use of the brevity penalty. In order to avoid punishing short sentences and to give some space for length variation, the BP is not applied on a per sentence basis but only to the whole document. A document level precision score for each n-gram is calculated as the geometric mean of the precision scores for the single sentences:

$$p_n = \frac{\sum_{C \in D} \sum_{w_1 \cdots w_n \in C} \text{clip\_count}_C(w_1 \cdots w_n)}{\sum_{C \in D} \sum_{w_1 \cdots w_n \in C} \text{count}_C(w_1 \cdots w_n)}$$

The final BLEU score combines the document level n-gram precision scores and incorporates the BP:

$$\text{BLEU} = \text{BP} \ \exp \left[\sum_{n=1}^{N} \frac{1}{N} \ \log \ p_n \right]$$

Papineni et al. (2001) found in their experiments that by using n-grams up to $n = 4$, the score correlates well with human judgment. The used weights for each n-gram precision are shown in the form the logarithm of the precision, weighted uniformly. This corresponds to a geometric mean, which performed best in the research of Papineni et al. (2001). Unigrams account for the content of the sentence and thus mainly present translation accuracy. Higher order n-grams instead account for the fluency of the translation.

As the BLEU score is widely used, it has also undergone more critical examination. Several issues have been reported: higher BLEU scores do not guarantee better translation quality (Callison-Burch et al., 2006); rule based MT systems in some cases get lower scores than SMT based systems despite higher human scores (Lee and Przybocki, 2005); BLEU scores work well on whole documents but have low agreement with human evaluation for single sentences.

Other Methods

Word Error Rate (WER) The word error rate is an edit distance measure that originates from the Levenshtein distance but uses words instead of characters as basic units. The WER measures the similarity of two sequences of words by evaluating the minimum number of deletions, insertions or substitutions needed to turn the candidate sentence into the reference. Another closely related measure is PER (position independent word error rate).

NIST As precision oriented measure closely related to the BLEU score is the NIST score (Doddington, 2002). This score improves the BLEU score by increasing weights of the more rare (harder to know) n-grams.

General Text Matcher (GTM) General Text Matcher is an automatic MT evaluation method proposed by Turian et al. (2003). Its goal is to provide a more intuitive automatic measure than the BLEU or NIST scores provide. The GTM measure is based on standard precision and recall combined to the

F-measure. These can be displayed graphically in an instinctive to interpret way. GTM allows evaluation of single sentences in contrast to BLEU and NIST scores, which evaluate sets of sentences.

METEOR METEOR (Metric for Evaluation of Translation with Explicit ORdering) combines unigram precision and recall as harmonic mean (Banerjee and Lavie, 2005). The authors created METEOR as improvement over BLEU, which also allows evaluating on sentence level. METEOR supports rule base or Wordnet stemming and Wordnet synonyms and seems to better correlate with human evaluation than BLEU and NIST scores (Koehn, 2007).

IQ_{MT} IQ_{MT} is a framework for evaluation that combines linguistic features on lexical, syntactic and semantic level (Giménez and Amigó, 2006). The comparison with human judgment (Koehn, 2007) indicates improvement over BLEU. Evaluation is possible on different levels and by combining different scores as BLEU, NIST, GTM, METEOR.

Whereas the first methods are computationally light due to their simplicity, the last two methods include additional components like stemming, synonym lookup or the combination of many other methods. This makes them computationally more intensive, which has to be kept in mind when large amount of data have to be evaluated in little time.

2.3 SMT Domain Adaptation

SMT adaptation aims to improve translation performance on specific domain text that is not pronounced in the bilingual training corpus. Domain adaptation is especially important in SMT systems, and has recently gained more research interest. As SMT systems are trained from empirical data, they are closely tied to the training data domain. Text corpora can be very different in many aspects (see Section 2.1.4), such as vocabulary, style or grammar. Therefore, the performance of SMT is more susceptible to domain differences than transfer systems, which are more independent of the corpus.

It is possible to improve in-domain performance without a dedicated in-domain bilingual corpus, as shown by the experiments of Ueffing et al. (2007a) as well as Wu et al. (2008). Ueffing et al. call their approach semi-supervised model adaptation, or transductive learning. Using the non-adapted system, they first translate a SL monolingual text corpus. Then, the good translations with high language model scores are selected and paired with their source

sentences to build a new synthetic in-domain corpus. By re-training the system with this corpus, valuable phrase table content will be strengthened, whereas the probabilities of useless content decrease. In this way the system gains knowledge from its own output.

A second approach without in-domain bilingual corpus is presented by Hildebrand et al. (2005). Their basic assumption is that the general language out-of-domain corpus is a compilation of different domain sub-corpora. Therefore, they filter the large bilingual out-of-domain corpus to select those sentence pairs only, which match the in-domain test set. In that way, the bilingual out-of-domain corpus is reduced to a bilingual in-domain corpus. The experiments for Spanish-English and Chinese-English language pairs show that their method results in a significant improvement in BLEU and NIST scores.

Along similar lines as Ueffing et al., the work of Wu et al. (2008) includes experiments, which use an in-domain source language corpus only, but their approach is slightly different. They also translate the SL monolingual in-domain corpus to create a synthetic bilingual in-domain corpus. But unlike Ueffing et al., they do not apply any filtering. Creating the synthetic bilingual corpus is repeated, iteratively improving the translation model, until the performance improves no further.

Another way to grow the bilingual in-domain corpus is to extract bilingual sentences from non-parallel corpora. This automatic creation of parallel corpora has shown to give promising results (Cheung and Fung, 2004; Munteanu and Marcu, 2006).

Xu et al. (2007) assume the availability of an in-domain bilingual corpus and describe an approach towards a multi-domain machine translation system. The different domain language models are combined as sentence level mixtures (Iyer and Ostendorf, 1996). Given K language models $P_k()$, the resulting interpolated model for a sentence $w_1, ..., w_i, ..., w_I$ is given by:

$$P(w_1, ..., w_i, ..., w_I) = \sum_{k=1}^{K} \lambda_k \left[\prod_{i=1}^{I} P_k(w_i|w_{i-1}) \right] \tag{2.13}$$

Different domain translation models are trained and optimized separately. During decoding, these models are combined as different features in a log-linear model (see Section 2.1.2). The feature weights are chosen on-line, depending on the domain of the input text. In their information retrieval approach, Xu et al. use a text similarity measure to choose the closest domain.

There are various approaches to language model adaptation, as enumerated by Béchet et al. (2004). Among others, he lists linear interpolation of out-of-domain and in-domain models, and a retrieval approach where documents matching the required domain are retrieved and trained on-line to create the in-domain language model. The work of Zhao et al. (2004) combines both of these approaches in the context of machine translation. Using the non-adapted system, they generate a list of translation hypotheses, which are used to create a retrieval query run against large-scale monolingual text corpora. The best result sentences are then used to train a new in-domain language model $P_A(w_i|h)$, which is linearly interpolated with the out-of-domain language model $P_B(w_i|h)$, using the interpolation weight λ:

$$\hat{P}(w_i|h) = \lambda P_i(w_1|h) + (1-\lambda)P_2(w_i|h) \tag{2.14}$$

Then, Zhao et al. run one more iteration: re-create the translation hypotheses using the interpolated language model, then build and run the queries and generate the in-domain language model. They achieve their best results using query models that incorporate additional structure in the queries.

Wu et al. (2008) use linear interpolation of language models as well as of translation models. However, instead of a given bilingual in-domain corpus, they employ an in-domain word dictionary for adaptation. This is done in different ways. One is to treat the dictionary as small in-domain phrase table, assigning uniform weights, constant weights or weights estimated by the translated monolingual in-domain corpus. Then, in-domain and out-of-domain phrase tables are combined during decoding. Either, each phrase table is used as factor in the log-linear translation model or both are linearly interpolated similar to the language models in Zhao et al., as shown in Equation 2.15:

$$P(t|s) = \lambda P_1(t|s) + (1-\lambda)P_2(t|s) \tag{2.15}$$

As intermediate results suggest that the log-linear approach works better, they use it in subsequent experiments. The other way to adapt is to add the dictionary to the bilingual out-of-domain corpus and train one large phrase table. Their experiments with Chinese-English and English-French language pairs show that all dictionary approaches improve the BLEU scores, with the dictionary-as-phrase-table methods outperforming the dictionary-concatenated-to-corpus approach. The best results are achieved when estimating dictionary weights from an in-domain corpus.

Wu et al. use two approaches for language model interpolation. One is the linear interpolation shown above (Equation 2.13) for $K = 2$, and the other one is to treat each language model as distinct factor in the log-linear

translation model (Section 2.1.2). Their experiments point out that linear interpolation performs better. They also combine all their techniques, dictionary, in-domain language model and the one described earlier (similar to transductive learning), which improved their baseline performance by about 8 BLEU points.

Koehn and Schroeder (2007) used a similar arrangement. Their simplest phrase table adaptation setup is to combine in-domain and out-of-domain bilingual corpora before training. A more advanced way is to create two separate phrase tables, which are combined using factored translation models (Koehn and Hoang, 2007). They create an adapted language model in different ways, either using only the in-domain LM, linearly interpolating it with the out-of-domain LM, or using both as separate features in the log-linear translation model. They experiment with 8 pairs of 5 European languages. Already using only an in-domain language model leads to significant improvements. Linear interpolation of language models achieves comparable results to using two separate features. Combining in-domain and out-of-domain corpora to train an adapted phrase table improves performance, but not as much as the factored model approach.

Brants et al. (2007) show that translation performance improves significantly which the size of the target language model. The translation scores continuously improve for their 5-gram language model when using between 13 million and 2 trillion tokens. For training language models of this size, they propose a distributed infrastructure. Given that the corpus size has to be doubled for each gain of about 2.5% BLEU, this methods interesting finding is of little practical value without the infrastructure to manage such huge amounts of data. The use of more effective methods for managing language models could mitigate this problem to some extent.

A quite different approach to domain adaptation is automatic post-editing (APE). In manual translation, a translator who corrects output from an MT system does post-editing. In automatic post-editing, the idea is that those corrections are used to train a system that automatically straightens out the translations. In such way, the post-edit system should relieve the editors of repeatedly fixing the same mistakes.

Isabelle et al. (2007) improve PORTAGE, a RBMT system, by the use of SMT as post-processing step. They work with the French-English translation pair and use a hand crafted correction corpus. A bilingual corpus is constructed using the RBMT output translations as source text and the post-editor reference translations as target text. This corpus is used for SMT model training. In this setup, two translation steps are performed: the source

text is translated by the RBMT to intermediate target language, which is translated by the APE layer to correct target language text. This process can be used to easily customize the RBMT system, or to adapt it to a specific domain. In their experiments, Isabelle et al. report results for a small APE-training corpus (<500k words) of human corrections. The system yields almost the same results in BLEU score, as an RBMT system customized with 18 000 manual entries. With a certain size of APE training data, the overall quality improvement stagnates. The improvements seem to be limited by the output quality of the RBMT system.

Simard et al. (2007b) report similar experiments using the PORTAGE system. Dugast et al. (2007) worked on improving the SYSTRAN RBMT system by statistical post-editing (SPE). They work with the English-French language pair and confirm good results by automatic evaluation as well as linguistic analysis. The SPE layer mostly improves local word choice, degrades morphological accuracy and does not affect long-distance reordering (which the RBMT does well).

In similar work, De Ilarraza et al. (2008) concentrated on the Spanish-Basque language pair, where little bilingual material is available. They use the open source RBMT system Matxin (Alegria et al., 2007). As Basque is a highly morphological language, each word in the source corpus was replaced by its stem and additional morphological tags. Tests with this morpheme-based SMT system show significant improvements in NIST, WER and PER scores over the word-based SMT system (except for BLEU scores, which are worse). Their results are consistent with other research for a restricted domain corpus. In contrary, for a general domain corpus the plain SMT system performs better than the combination of RMBT system with SPE module.

Other work regarding domain adaptation includes studies about using mixture models for SMT (Civera and Juan, 2007; Foster and Kuhn, 2007).

This chapter contained a review of the most important theoretical concepts for the experiments that were conducted as part of this thesis. The next chapter explains these experiments in more detail.

Chapter 3

Experiments in Domain Adaptation

This chapter contains a detailed description of the conducted experiments, which includes:

Preparation contains details of the baseline system, the in-domain corpus, the collected feedback data, users and the feedback system itself.

User Feedback Collection describes how the baseline system feedback and reference translations for the news domain corpus were collected.

Adaptation models describe which approaches were used to improve the baseline system for the news domain.

Adaptation Systems Training describes how the model adaptation was performed.

Evaluation outlines the methodology used for automatic judgment of adaptation performance and system ranking.

Originally, a second round of user feedback collection was planned as last step. Volunteers would have rated the quality of adapted translation models in order to validate the automatic evaluation measures. This human evaluation effort was not conducted, due to time constraints.

3.1 Preparation

The main goal of this thesis was to conduct experiments in domain adaptation of statistical machine translation. To make that possible, a number of things were required:

- A bilingual corpus, used to train the baseline SMT system and software for training the models and carrying out the translation (decoding).
- A bilingual in-domain corpus, which translates badly using the baseline SMT system.
- Automated and human methods for assessment of the translation quality before and after domain adaptation.
- Subjects who perform the human assessment.
- Software that is used for the assessment.

The following explains how these requirements are complied with.

3.1.1 Baseline SMT System

The baseline SMT system was created employing a large bilingual English–Finnish text corpus to train Moses (Koehn et al., 2007), a statistical machine translation system, in two different ways: the first branch of the baseline uses words as basic units and the second one uses morphs (Section 2.1.3). The Moses software is also used to carry out the translations. A decision was made to use the baseline system from existing work in our laboratory (Virpioja et al., 2007). For the user feedback collection, the translation models from the mentioned studies were used, whereas the models for the domain adaptation were created again. The baseline system uses a 4-gram language model, which was created using the SRILM toolkit (Stolcke, 2002). The used reordering model was of the default type `msd-bidirectional-fe`. For the user feedback collection, the translation table and reordering table were filtered (Johnson et al., 2007); this was not done for the domain adaptation experiments.

Bilingual Corpus

The baseline models were generated from the Europarl corpus version 2 (Koehn, 2005), which is a widely used parallel corpus in statistical ma-

chine translation. The corpus is freely available and based on the web versions of the European Parliament proceedings from April 1996 to September 2003. The corpus contains text in eleven European languages (Danish, Dutch, English, Finnish, French, German, Greek, Italian, Portuguese, Spanish, Swedish) with about 20 million words per language. Although the newer version 3 of the corpus is available since September 2007, adding another 3 years of proceedings data, the older version was used in the experiments so that available translation models could be re-used.

For the experiments in this thesis, we selected the English–Finnish data. It contains the proceedings data from January 1997 to September 2003 with a total of 1.3 million bilingual aligned sentence pairs before, and 0.8 million after pre-processing. The raw corpus contains a total of 45 million words.

The pre-processing consisted of sentence boundary detection, tokenization, sentence alignment, noise removal and long sentence removal. For sentence boundary detection, regular expression rules and language dependent abbreviation lists were used. Sentence alignment was done using the algorithm described by Gale and Church (1993), which uses sentence length to find corresponding sentences. Noise, such as special characters, was removed and the text was lowercased. Long sentences cause computational problems during statistical word alignment, therefore sentences longer than 100 words were removed, which removed about 400 000 sentence pairs.

Table 3.1: Number of sentences, distinct words, total words and type to token ratio for the unprocessed (raw) and pre-processed (pp) parallel Europarl corpus.

language	type	sentences	word tokens	word types	characters in million	type-token ratio
Finnish	raw	1 262 914	18 837 151	479 779	146	0.0255
English	raw	1 262 914	26 073 619	83 496	143	0.0032
Finnish	pp	865 732	17 183 927	455 359	133	0.0265
English	pp	865 732	23 863 424	78 944	131	0.0033

Some further corpus statistics can be seen in Table 3.1. The raw corpus is aligned using the alignment script from the Europarl project, but otherwise unprocessed. Note some characteristic features of a highly inflected language like Finnish. Each word can comprise several concepts as pre- or postfixes, which other languages instead express as separate words, such as prepositions. This, as well as compound words, cause a lower amount of total word

tokens and a higher amount of distinct word types, when compared to the English part of the corpus.

A text example from the corpus can be found in Appendix A.1 in Table A.1. Koehn (2005) contains further details related to the Europarl corpus, in particular about collection of the data, pre-processing and alignment.

Software

The open source statistical machine translation toolkit Moses (Koehn et al., 2007) was used to train the baseline and adapted models and to translate text. Training models with Moses is a process, which includes several steps as shown in the following:

1. **Preprocessing of the corpus:** each word is assigned a numerical identifier and the word tokens in the corpora are translated into numbers.
2. **Finding word alignments:** using the open source GIZA++ toolkit (Och and Ney, 2003), two word alignment files for both directions are extracted, source to target and target to source. GIZA++ implements, among others, the IBM models 4 and 5 as well as an alignment model based on word classes, which are found by the mkcls tool (Och, 1999).
3. **Finding phrase translations:** a number of heuristics are used to build phrase alignments from the word alignment files.
4. **Scoring every extracted phrase translation:** the five different features explained in Section 2.1.2 are used.

The Moses toolkit also comes with a decoder that uses the created models to find a probable target translation for a source sentence. Beam search, a heuristic search algorithm, is used to shorten the search time at the cost of an optimal solution. The different models are interpolated in a log-linear approach as outlined in Section 2.1.2.

Word Based Models

Our main models were Moses models, trained from the word tokens in the pre-processed bilingual corpus. The resulting models were called 'word-based models', and used for user feedback collection and domain adaptation experiments.

Morph Based Models

A second model, the 'morph-based model', was used for user feedback collection. The Morfessor (Creutz and Lagus, 2007) software was used to segment words into morphs, which are minimal meaning bearing units of the language. Morfessor learns morphology in an unsupervised manner and was trained on the Europarl corpus. Further details can be found in section 2.1.3. The following Table 3.2 shows an example of the translation process including segmentation. The Finnish word-based sentence (1) is segmented into morphemes (2), then translated into English morphemes (3) and combined to English words (4). The tags after the morphs indicate if the morph is a prefix (PRE), stem (STM) or suffix (SUF).

Table 3.2: Examples of the morph-based translation process. The Finnish word-based sentence (1) is segmented into morphemes (2), then translated into English morphemes (3) and combined to English words (4). Stems are printed in bold, suffices in italic.

1	saksalainen		kritisoi	voimakkaasti	olosuhteita	.
2	**saksa**+**lainen**		**kritisoi**	**voimakkaa**+*sti*	**olo**+**suhteita**	**.**
3	**the**	**german**	**strong**+*ly*	**criticise**+*d*	**condition**+*s*	**.**
4	the	german	strongly	criticised	conditions	.

The segmentation and combination are done as pre- and post-processing. Moses is trained with the morpheme representation of the words resulting in the morph-based models. The translation step uses these models.

Finding good word alignments is more challenging for morphologically rich languages. As can be seen in the type-token ratio in Table 3.3, Finnish has a much higher value having less word tokens in total and more word types. Word alignment is harder to learn for a word type that occurs less often, or in few different possible contexts. As each type occurs less frequently, there is less structural evidence, and highly inflected word types hide similarities between words. A larger corpus could alleviate this, but is not available.

It is reasonable therefore to segment words into morphs, which have independent meaning and represent different concepts. This allows preserving the closeness of different word types when they share common morphs. Segmentation into morphs also helps to translate unseen words by splitting them into their constituents. If the constituents can be translated independently, the result has some chance to be a proper translation or at least to be understood. Morphological segmentation has shown some potential for improved translation results (Maucec et al., 2006; Bojar, 2007).

The morphological segmentation and Moses models had been trained in a previous work by Virpioja et al. (2007). Using BLEU score automatic evaluation, they could not yet show that the morph models yield better translation results. That was one reason to include the models into the experiments of this thesis: to validate the BLEU measures by human evaluation. Maybe the morph models result in better-perceived translations that do not correlate with plain BLEU measures.

Table 3.3: Number of sentences, distinct words, total words and type to token ratio for the pre-processed parallel Europarl corpora. The first part is for the word based corpus; the second part is for the morph-based corpus.

language	sentences	word tokens	word types	characters in million	type-token ratio
Finnish word	865 732	17 183 927	455 359	133	0.0265
English word	865 732	23 863 424	78 944	131	0.0033
Finnish morph	857 892	27 040 843	78 104	264	0.0029
English morph	857 892	27 534 433	45 956	265	0.0017

The Morfessor software suite provides several different models. In our experiment, the latest model, Categories-MAP was used. Further details about the functioning of Morfessor can be found in Section 2.1.3.

3.1.2 In-domain Corpus

The monolingual source language in-domain corpus was extracted from the web version of Iltalehti, a Finnish daily tabloid newspaper. The corpus statistics are shown in Table 3.4.

Table 3.4: Number of sentences, distinct words, total words and type to token ratio for the Iltalehti corpus.

language	sentences	word tokens	word types	characters in million	type-token ratio
Finnish word	72 128	886 678	148 878	6.8	0.17

Preprocessing

As the in-domain corpus was collected by other fellow researchers, it had already some pre-processing applied:

- Every line contained one sentence.
- All words were lower-cased.
- Commas were replaced with the special token 'C'.
- The sentences ended with the full stop '.' as an own proper token.

The sentence length was limited for our purposes to minimum 3 and maximum 12 words. Shorter sequences than 3 words were not considered proper sentences and sequences of more than 12 words were considered complex and too laborious to manually evaluate and correct.

Quite a number of sentences contained nonsense. A large part of them was filtered out by applying simple rules. Some examples of corrupt sentences are shown in Table 3.5.

Table 3.5: Examples of corrupt sentences from the in-domain corpus.

525 l l tämä tuli mieleeni C kun luin tom westergårdin haastattelun .
tuhanteenyhdeksäänsataanseitsemäänkymmeneenkuuteen
quite a lot that start with a comma
C joka on leinon runoista kuuluisin ja rakastetuin .
or with a (with was only once meaningful for me 'a la carte..')
a yli kaksi tuntia .
or with hhh ?
hhh kolme jursinov ja tps ovat jo todistaneet taitonsa .

Comparison of Out-of-domain and In-domain Test Corpora

To compare out-of-domain and in-domain corpora we conducted what Rayson (2003) calls a Type A comparison. That means the comparison of a sample corpus with a large standard corpus. The large corpus is referred as the normative corpus.

Using the log-likelihood ratio measure presented in Section 2.1.4, we analyzed, which word types are distinctive for each corpus, i.e. the word type count difference between the two corpora is statistically significant at the

95% level (p-value 0.05). A subset of the resulting ranking of characteristic word types for each corpus is shown in the Appendix A.2 in Table A.2 and Table A.3. The difference in corpora becomes apparent by looking at the word lists. Table 3.6 shows a summary. Distinctive for the Europarl corpus are words used in European Union Proceedings and words that are more often used in longer sentences. Distinctive for the Iltalehti corpus are typical news text words, Finnish given names as well as other local words.

Table 3.6: Distinctive word categories after a domain comparison between Europarl and Iltalehti corpora

Europarl	Iltalehti
European commission typical: europaan, puhemies, komission komissio, parlamentin, unionin, unioni, neuvoston, parlamentti, jäsenvaltioiden, esittelijä, mietintö, mietinnössä	***Finnish given names:*** mika, kari, juha, pekka, jari, matti, jukka, paavo, antti, janne, mikko, lola, ari
More likely in longer sentences: että, jotka, täme, tämän, ja, jotta, tätä, tässä, sen, tästä, nämä	***Typical news words:*** poliisi, mm , elokuva, tv, ollut, tuli
	Colloquial language: mä

Selection of Evaluation Translations

We selected 1 000 sentences randomly from the in-domain corpus. Table 3.7 shows the sentence length distribution, which is, as required, close to uniform. The sentences were translated with the word baseline system and with the morph baseline system giving a total of 2 000 sentence pairs.

3.1.3 Collected Feedback Data

We selected the set of collected feedback data by having the goals of the experimental design in mind:

1. Gather in-domain reference translations of Iltalehti source sentences to be able to train adaptation models and to evaluate the translation quality.

2. Confirm if the morph-model achieves higher human ratings.

Table 3.7: Distribution of sentence lengths of the chosen sentences from the in-domain corpus. The length of sentences is given in words, the tokens ',' and '.' were not included.

sentence length	count	percentage
4	3 312	8.75
5	4 533	11.98
6	4 990	13.19
7	5 191	13.72
8	5 155	13.62
9	5 063	13.38
10	4 933	13.04
11	4 663	12.32

Translation Quality Measures

Human evaluation was chosen as primary evaluation method and to evaluate automatic measures. The collected scores were used to assess the quality of the baseline translations after user feedback collection (Section 3.2), as well as to assess the quality of the adapted translations in a second user feedback round. Due to time constraints, this second round could not be conducted.

Intelligibility Intelligibility is the ease with which a translation can be understood. This measure is one of the most distinctive features for translation quality and was the first part of our human evaluation. A 5-point scale was adapted from Trujillo (1999), which was originally suggested by (Nagao et al., 1985). We used a 5-point scale for brevity reasons to limit space requirement in the evaluation tool and to reduce reading time. Additionally the verbose descriptions were shortened. User tests showed that people did not understand the scale because they had not read it thoroughly. Thus the scale descriptions were considered too complicated and were shortened to build a concise bullet list for each rating.

A second user feedback test revealed that it was hard to choose between ratings for some groups of sentences. This happened especially in the middle of the scale (between ratings 2 and 3). Using this feedback, the descriptions were rewritten to minimize ambiguity and simplify assignment of ratings to sentences. The used scale can be found in Table 3.8. We did not follow any formal process to construct a methodologically sound scale.

Accuracy Accuracy measures to what extent the translation conveys the same meaning as the original sentence. As the intelligibility scale, our 5-point accuracy scale was adapted from the one presented by Trujillo (1999). With the help of user feedback, conciseness and ease of assignment were iteratively improved. The used scale is shown in Table 3.9. Similar to the intelligibility scale, also the accuracy scale was created using a best effort method.

Corrected Text

The last part of the data, which was gathered from the users, were corrected translations. The correction for a translation was defined as "what you think is the best (British) English translation for the original sentence".

One can distinguish between reference translation and correction. The reference translation would be the best possible translation (there might be several) whereas the correction would be the correct sentence that can be easily made out of the given candidate translation. This difference was not explicitly pointed out to the users, although the description suggests that the reference translation is desired. What the best translation is, is explicitly left to the judgment of the user. One possible definition is the translation that best conveys the meaning of the original sentence. The users were asked to correctly write punctuation as well as proper capitalization like in names or abbreviations.

Time Measures

Additionally to the already given data, the time used for rating and correction was recorded. This was done for usability reasons, so that rating and

Table 3.8: Used intelligibility scale, measured from 1 (worst) to 5 (best). The descriptions were made to be quick to read and easy to understand.

Rating	Description
5	clear meaning (correct grammar)
4	almost clear meaning (small mistakes)
3	the meaning can be guessed (parts are understandable)
2	the meaning can hardly be guessed (parts are understandable, but worse than above)
1	nothing understandable

Table 3.9: Used accuracy scale, measured from 1 (worst) to 5 (best). The descriptions were made to be quick to read and easy to understand.

Rating	Description
5	content faithfully conveyed (no changes required)
4	content almost faithfully conveyed (minor changes required)
3	parts of the content conveyed (some changes required)
2	content not adequately conveyed (major changes required)
1	content not conveyed at all (complete rewrite required)

correction times could be optimized. Other uses could be imagined, but were out of scope for this work. Additionally the data is expected to contain many outliers due to the nature of the application setup where users could have breaks during the feedback.

Required Data Amount

The basic idea of collecting the feedback data is to obtain a test suite for the in-domain corpus. As it is not trivial to estimate how much data would be required for successful domain adaptation, the setup of comparable MT evaluations was examined. NIST MT Eval, make use of test corpora made of several hundreds of sentences. Given that figure we set our goal to 1 000 sentences (actually 1 000 source sentences with two different translations each, which gives 1 000 sentences to be translated and 2 000 translation pairs to be rated). We estimated that every user would rate and correct about 15 sentences, which would require a total of about 130 users to obtain the 2 000 translations.

Automatic evaluation measures correlate best with human evaluation when several translations exist which cover the translation variability well. Some setups suggest up to five reference translations. Here we are in a dilemma: do we need several reference translations at a cost of the total number of translations? Our trade-off is one reference translation for each translation pair, which results in two reference translations for each source sentence.

3.1.4 Users

As estimated in the previous section, a planned prerequisite of 130 users would be a sufficient amount for the experiments. No funding was available

for hiring users. Anyhow, that would not have been in the spirit of the overall vision, which is based on voluntary users who contribute to improve the system.

Participating users needed to have adequate English and Finnish skills (over 95% were native Finnish speakers). English skills were not verified, as it was already hard to find enough participants. However, random examination shows that the given corrections are of good to very good quality.

3.1.5 Feedback System

The feedback system is the software used for the human assessment. The two essential functional requirements are:

- Users can evaluate translations by rating intelligibility and accuracy.
- Users can give a correction to existing translations.

Besides these, some non-functional requirements were identified. The application should:

- have a clean and smooth user interface to avoid irritating users.
- have built-in motivational factor to incite users to participate.
- provide reasonable security for the gathered data.
- support current versions of Mozilla based browsers, Safari and Internet Explorer.
- be reusable to some extent for later efforts of building a web-based machine translation community.

Decisions

A typical web based application can serve the requirements mentioned in the previous section. Using a personal account for each user including log-in credentials satisfies the security requirement. In this way, any collected data can be linked to its user allowing easy removal of spam (improved security) and the creation of a high score list (a motivational factor). In addition, it enables further user behavior analysis. Creating a personal account is laborious, which prevents people from taking a quick glance at the application. This drawback was accepted.

The use of client side scripting with JavaScript places additional requirements to the client browser but drastically improves application interactivity and usability.

In order to reuse parts of the created application, a web-application framework was used to allow easier scaling of the application. In order to reuse the collected data, a relational database was employed to structure the data and to allow further flexible data processing. The, typical for a web application, concurrent data access is simplified by using a database.

Used Software Python, a high-level, object-oriented and open programming language, was chosen for the implementation. First experiments with a simple python CGI showed that the required complexity demands a more structured approach. Therefore a web application framework was selected. Django (Django Software Foundation, 2008) was chosen as a high-level python web framework encouraging "rapid development and clean, pragmatic design". Although the learning curve was not flat, the choice turned out to be a good one. As database management system, MySQL (MySQL AB, 2008) was selected due to existing skills and its open availability. The applications were located on a Debian Linux server.

User Interface Decisions A number of decisions were made in favor of keeping users motivated instead of trying to get accurate results. The motivation behind this was the worry about getting too little feedback in total.

After getting testers' feedback on a first prototype, rating translations and correcting them was divided in a more clear 3-step process. First, the user only sees the English sentence and rates its intelligibility. Then only, the Finnish source sentence appears as well, so that translation accuracy can be rated. Having both sentences shown at once would influence the intelligibility rating. As next step, the translation pair is shown and the user has to input a correct translation (or copy the given one). Untranslated words (the translation system did not know them) were hidden behind a black box in the rating part, so that they could not help users to understand the sentence meaning, and therefore influence the rating. However, for motivational reasons, this was chosen to be a soft restriction: by moving the mouse pointer on top of the black box, users could peep to see the untranslated word. The instructions told that unknown words, which were not people or place names, lower the rating. This design choice might bias the intelligibility results towards better ratings.

Other features The application contained a help page where the rating sentences could be trained. This was not mandatory for users, but one means to improve inter-rater reliability. In order to provide a correct English translation, the user could pop up a third-party Finnish-English dictionary or run a spell-check on the input. A high score page showed the 10 users with most translations. Their translations including rating could be browsed as a mean of motivation as well as preventing people from giving nonsense corrections. A total translation counter showed the progress of all users. Finally, users could send comments about the application.

Screenshots of the web application as well as the data model can be found in B.2.

3.2 User Feedback Collection

We sent invitation email to friends and colleagues to obtain volunteers (see Appendix B.1). The total amount of participating users was 30, although at least 100 users received the email. Users who went to the evaluation web page, had to first create a log-in account. This way was used to be able to personalize feedback input for later analysis, to reduce abuse of the system (spam) and to add an incentive in the form of a little contest using a high score list showing the 10 best contributors.

The users then chose to start a feedback round where they assessed machine translations and corrected them. It was possible to choose a set of either 10 or 20 sentences. There was nothing besides self-discipline that enforces the completion of the round. Any amount of rounds was allowed.

It was optional to read an introduction that explained the terms and feedback process. This part also contained examples for the intelligibility and accuracy measures in order to ease the rating decisions during the feedback. Although reading the introduction would have shortened the adaptation phase, it was not mandatory in order to keep the motivation high and prevent driving away users before they had even started.

In the feedback itself, intelligibility was rated first. The translated English sentence was shown together with the five point scale as described earlier Section 3.8.

3.3 Adaptation Models

The review of existing work on SMT domain adaptation (see Section 2.3) reveals many promising and interesting approaches. Given the collected Finnish-English bilingual news domain corpus, the work that uses in-domain bilingual material is naturally closer. We decided to perform experiments in 4 different families:

1. Adaptation of the language model in order to confirm the best alternative, which is subsequently used in the other families.
2. Adaptation of the phrase table via the concatenation of small in-domain to large out-of-domain bilingual corpus.
3. Adaptation of the phrase table via log-linear interpolation.
4. Adaptation of the system via a post-editing module.

For the interpolated phrase table and the post-edit module, a variation using an out-of-domain word alignment dictionary was studied. These experiments will give an idea about how well the domain of the Europarl baseline system (out-of-domain) can be adapted to the collected Iltalehti news domain corpus (in-domain). With respect to the used Finnish-English language pair, it will be interesting to contrast our findings with previous work. Below we introduce family and experiment identifiers to simplify the result tables and figures. We also use the abbreviations for translation model (TM), reordering model (RM) and language model (LM) here and in the results section. Depending on the corpus used for training ('ep' for Europarl, 'il' for Iltalehti, 'pec' for post-edit corpus), these can get different values.

3.3.1 Language Model Adaptation

The experiments in the language model adaptation family were prefixed with the letter *L*. The baseline system uses the out-of-domain (Europarl) language model only. We tried 3 different approaches to adapt the language model to the new domain (Iltalehti). In experiment *L1*, the target language parts of the bilingual in-domain corpus and the out-of-domain corpus were concatenated (LM: ep+il). Experiment *L2* uses two different language models, one general and much larger out-of-domain language model and one quite small in-domain language model. The latter is trained in the same way as the

out-of-domain language model. Both are used as different factors in the log-linear translation model (see Section 2.1.2), and are in that way interpolated (LM: ep, il). As it was decided to omit model tuning, the language model weights were not optimized. Preliminary experiments showed that too much weight on the in-domain model strongly degraded the performance. Therefore it was decided to give the in-domain model only slightly more weight (0.6) than the out-of-domain model (0.4). Koehn and Schroeder (2007) use similar weight values in their experiments. The third experiment, *L3* uses the same in-domain language model as *L2* but without adding the out-of-domain language model (LM: il). In all these experiment, the translation model and reordering model are the same as the baseline (TM: ep and RM: ep).

3.3.2 Concatenate Model

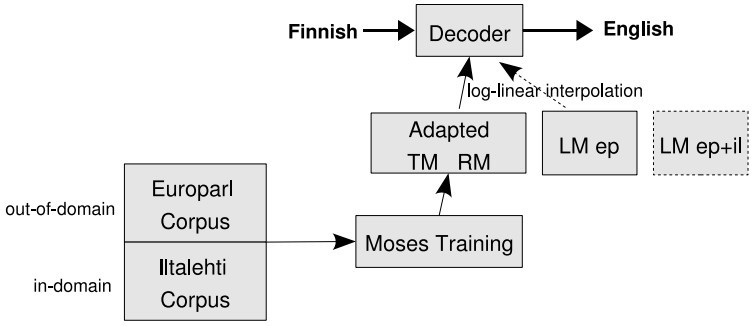


Figure 3.1: In the Concatenate Model, in-domain and out-of-domain corpora are concatenated prior to model training.

In the concatenate model family (prefixed by *C*), we use the in-domain data as additional data to the baseline system (Figure 3.1) Thus, the bilingual in-domain corpus is simply concatenated to the out-of-domain data. Then, the translation model (TM: ep+il) and the reordering model (RM: ep+il) are trained. Two experiments were conducted here, one with (LM: ep+il) and one without adapted language model (LM: ep). This is displayed in the figure as two LM options, but only one at a time is used. As preliminary tests showed, experiment *L1* outperformed *L2*. Therefore, we simply use the *L1* "ep+il" approach for further language model adaptation.

3.3.3 Interpolate Model

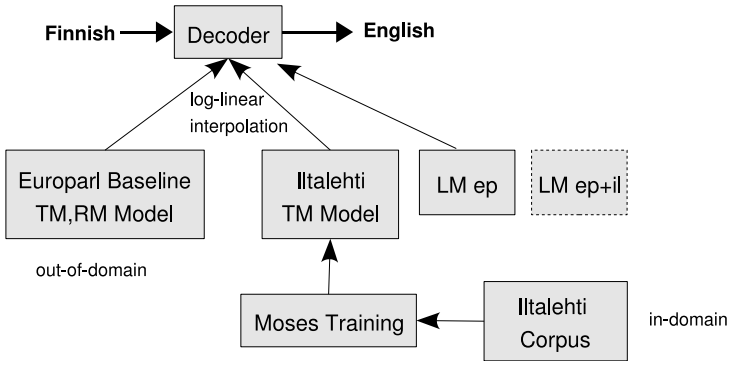


Figure 3.2: In the interpolate model, log-linear interpolation is used to combine several translation models.

In the interpolate model family (prefixed by *I*), two distinct translation models are used: the existing baseline model and an in-domain model (Figure 3.2). Then during decoding, both are interpolated as different factors in the log-linear translation model (TM: ep, il). Note that no separate in-domain reordering model was used (RM: ep). Again, two different language model options were used. *I1* is the one without adapted language model (LM: ep) and *I2* the one with adapted language model (LM: ep+il).

As it seemed that the in-domain corpus is rather small for the used statistical word alignment to work properly (only about 900 sentences), we used a setup to boost the word alignment. The Europarl word alignment dictionary was filtered so that only those words were left, which were included also in the bilingual in-domain corpus. Then this dictionary was added to the in-domain corpus and used during GIZA++ word alignment. Afterwards, the parts used for boosting were removed again so that only the in-domain data was put forward to the further training steps (TM: ep, il+d). The interpolated dictionary approach without adapted language model is called *I3* and the one with adapted language model *I4*.

3.3.4 Post-edit Model

The setup of the post-edit family (prefixed by *P*) is most different from the previous approaches, as it requires two decoding steps. The first one uses the baseline system. The output translations are considered to need correction,

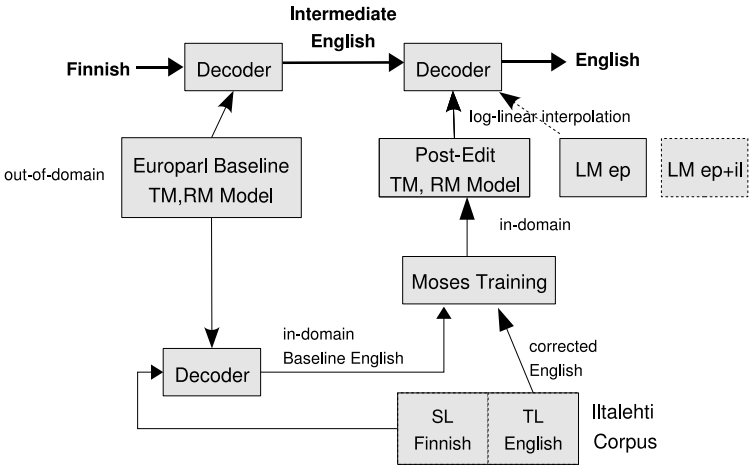


Figure 3.3: In the post-edit model, a separate translation layer is created to simulate human post-editors.

and are therefore translated again using the post-edit model to yield better translations.

For the training of the post-edit model, the out-of-domain baseline model is used to generate translations of the in-domain source language (SL) monolingual corpus. These translations will be paired with the in-domain target language part (the corrected English) to form the monolingual post-edit training corpus (which we will call 'pec').

Our first experiment (*P1*) uses the translation and reordering model trained from the described corpus (TM: pec RM: pec), and keeps the out-of-domain language model (LM: ep). The same setup but applying the adapted language model (LM: ep+il) is called *P2*.

3.4 Adaptation Systems Training

Training the adaptation models was done on a Linux cluster. The nodes used were IBM eServer 325 with 2*AMD Opteron 248 CPU's (2.2GHz, 1MB L2 Cache) and 4GB RAM. For training and decoding, the moses build from 11.12.2007 was used throughout all experiments. BLEU evaluation was performed with the script included in moses (multi-bleu.perl, from 14.03.2007). For all experiments, the same versions of the scripts were used.

Training times depended on the cluster load, but the concatenate models

took the longest with about 2.5 days (for about 800 000 sentence pairs). Interpolate and post-edit models trained in about an hour. Given our 10 fold-cross validation approach, 10 models for each family had to be created. The time required for evaluation of one experiment (translation of the test corpus, BLEU scoring, bootstrap resampling) was about 2.5 hours.

In order to be able to compare the different experiments, it is important that parameters are kept as similar as possible. This posed a problem across experiment families. If one wants to determine which one of two families is better, the best version for each family should be subject of comparison. However, as no model tuning was performed, no best version was available. The default parameters could be much worse than optimized ones. Without tuning, the real performance potential of an experiment family remains hidden. Despite of these issues, the decoding parameters (moses.ini) were kept as similar as possible across all experiments.

3.5 Evaluation

When comparing the performance of different MT systems, it is important to know which methodology is used to derive the results. The choice of a different methodology could turn any result. Therefore, the results have to be seen in the context of the used methods.

3.5.1 Choice of Automatic Measures

We chose to use the BLEU score measure. One reason is that it is almost standard in most MT literature nowadays, despite many critics against it. Originally, we had planned to include more advanced automatic evaluation measures like METEOR or IQ_{MT} to allow for comparison with human evaluation. Due to change in focus, space restrictions in the result tables and simplicity, we only use the BLEU measure.

3.5.2 Improving Statistical Accuracy

Given our small bilingual in-domain corpus, it is hard to obtain a representative sample and the statistical significance of our results could be questioned. In addition, it is impossible to report a confidence interval of some statistic having only one sample. Therefore, a method was used that allowed us to

report confidence intervals and improve statistical accuracy. Bootstrap resampling would be such a method (Efron and Tibshirani, 1986). However, given that for each resampled in-domain corpus, one translation model has to be trained, exhausts our computational resources (thinking about 100 systems, each of which takes 2.5 days to train). Therefore we chose a combined 10 fold cross-validation and bootstrap resampling approach. In that way, cross-validation enlarges the variability in training sentences and bootstrap resampling improves statistical accuracy for test set evaluation.

Using 10 fold cross-validation, the collected Iltalehti corpus of 1 076 sentences was split into 10 non-overlapping sets of about 968 training sentences and about 108 test sentences. As the total amount of in-domain data is rather small, a separate validation set for tuning was not chosen. Doing so, would have decreased the amount of data used for training and testing too much. The training parts were used to train translation models, reordering models and language models. The testing part was only used during evaluation for automatic scoring.

Bootstrap Resampling

In order to get a confidence interval for the automatic scores, the target language translations of each cross-validation model were resampled without replacement to form 1 000 new sets of 100 sentence test corpora. This method is known as bootstrap resampling and has been applied in the context of significance tests for machine translation by Koehn (2004) and in little variation by Zhang et al. (2004). The basic assumption behind bootstrap resampling is stated by Koehn as: "Estimating the confidence interval from a large number of test sets with n test sentences drawn from a set of n test sentences with replacement is as good as estimating the confidence interval for test sets size n from a large number of test sets with n test sentences drawn from an infinite set of test sentences.". The advantage of the approach is that only a small amount of test sentences have to be translated, or in our case that it is sufficient to only have a small amount available.

As described above, applying the bootstrap method for each cross-validation model yielded 1 000 evaluation scores. For final evaluation, these $10 \cdot 1\,000$ scores were combined to determine the confidence interval. The same data set was used to test whether one model scores significantly higher at the 95% confidence level. We calculate the confidence interval as described in Zhang et al. (2004), but use a 95% one sided interval. For comparing two system, we perform a pairwise comparison. The method can be described as

first calculating the difference between each paired sample and subsequently verifying if 95% of the differences are larger zero for any one of the participating systems. If the condition is met, the score difference is significant at the 95% level. To our knowledge, this procedure has not been widely applied in MT research yet, therefore we also rank the systems using the Wilcoxon signed rank test (Wilcoxon, 1945) and Student's t-test for comparison. In our opinion, a non-parametric test like Wilcoxon's signed rank test is more appropriate than the t-test, as we cannot assume the data to be normally distributed. A large part of the MT research uses Student's t-test for significance testing, therefore we include it for comparison. Preliminary results showed that the bootstrap method is much more pessimistic in reporting a statistical significance between two systems than the Wilcoxon signed rank test and Student's t-test. As Riezler and Maxwell (2005) describe the bootstrap method as too optimistic, we use it over the other two, even less strict methods for system ranking.

Reported Measures

For the bootstrap data, the confidence interval as well as the mean (both in BLEU points from 0 to 100) and relative standard deviation (RSD) are reported (in %). RSD, which is a precision measure, is used to better compare the range of the scores across systems. Besides the bootstrap measures, we also report the means of the 10 cross-validation results and the score of the training evaluation. For the training evaluation, all available data from the bilingual in-domain corpus (that is training and test set) were used for training and evaluation. These were reported for comparison mainly and to see, if a system at least memoizes the trained sentences.

The systems were ranked using the above-mentioned statistical tests: bootstrap resampling, Wilcoxon signed rank and Student's t-test. In addition, a histogram of the bootstrap values is presented to allow for visual comparison of the results.

Chapter 4

Results

This results chapter shows the observations of the experiments. The first section contains results of the user feedback collection. The results of the adaptation experiments are presented in the second section.

4.1 Analysis of User Feedback Data

Feedback data is all the data that was collected by users during the user feedback collection (Section 3.1.3). Here we present statistics about the amount of feedback, time measures and translation quality measures.

4.1.1 Amount of Data

Table 4.1 shows the amount of collected feedback, with a total of 1 074 feedback ratings, which means that 54% of the 2 000 planned ratings (see Section 3.1.3) were achieved. We obtained 93% of the planned amount of translations

Table 4.1: Amount of feedback received for sentences, categorized by model (word and morph) and source sentence translation.

	Word Model Ratings	Morph Model Ratings	Total Ratings	Source Translations
available	1 000	1 000	2 000	1 000
collected feedback	510	564	1 074	930
coverage	51%	56%	54%	93%

(930), however only 14% of the sentences got two translations. The amount of received word and morph translations is roughly equal (54% and 56%), as expected.

The feedback accumulated during a period of 20 weeks, although the main contribution took about 12 weeks. The accumulated amount of feedback can be seen in Figure 4.1. The total number of users that supplied feedback was

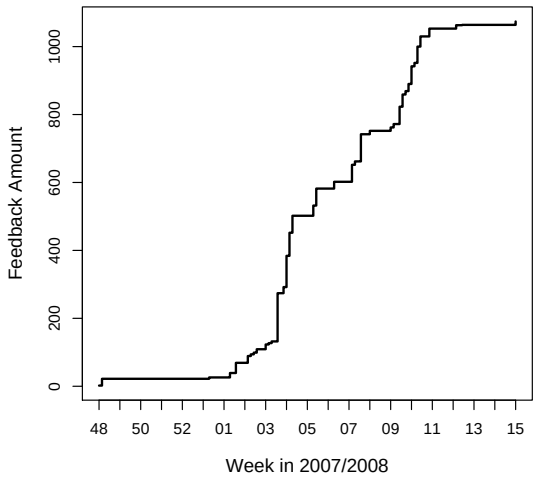


Figure 4.1: Accumulation of user feedback data over time.

relatively low. Only 25 people created an account and 21 of them contributed data. Table 4.2 splits users into groups based on their contribution. The amount of feedback per user was relatively high: 43% of the users rated and corrected 60 and more translations.

Users were able to mark source sentences as bad, which means that they are not proper sentences to be translated. The users marked 77 sentences as bad. Some representative example sentences are shown in Table 4.3.

Table 4.2: User contribution shown by how many users have given which amount of feedback.

Feedback Amount	0-10	11-25	26-59	60-95	96-115	115-158
Users in that range	6	4	2	4	2	3

Table 4.3: Examples of source sentences that were marked as bad sentence by the volunteers.

C vanilja tai kookos .
chill out duo pine apple circlen täyspitkä cd distant adrifting circles .
ajax juventus on unelmafinaali .
honey moon on the rocks .
i jala samu , i jala samu , joo .
nolla , tommi anonen nolla .

4.1.2 Time Measures

For each user's feedback, the time taken for rating a sentence as well as correcting the translation was recorded. The summary statistics are shown in Table 4.4.

Table 4.4: All users' summary statistics of rating and correction durations in seconds. Given are minimum, lower quartile Q_1, median Q_2, upper quartile Q_3, maximum, mean $\bar{x}$ and standard deviation s.

	min	Q_1	Q_2	Q_3	max	$\bar{x}$	s
rating duration	1	20	30	50	979	56.38	93.9
correction duration	3	33	62	131	3 089	130.8	232.6

Summaries of the rating- and correction durations are shown on a per user basis in Figure 4.2 and 4.3 respectively. For each user, the figures show one box-plot (also called box-and-whisker diagram), which denotes minimum non-outlier value, lower quartile, median, upper quartile and the maximum non-outlier. The box represents the inter-quartile range, which contains 50% of all values for that user. For the box-plots, the largest outliers were removed. Those were 9 values for the rating duration (up to a maximum of 19 hours), and 11 values for the correction duration (up to a maximum of 51 minutes). People who interrupted the feedback while doing something else had probably caused these outliers. Any remaining outliers are shown in the box-plots as circles. The figures show that some users spent considerably more time for rating and correction than others.

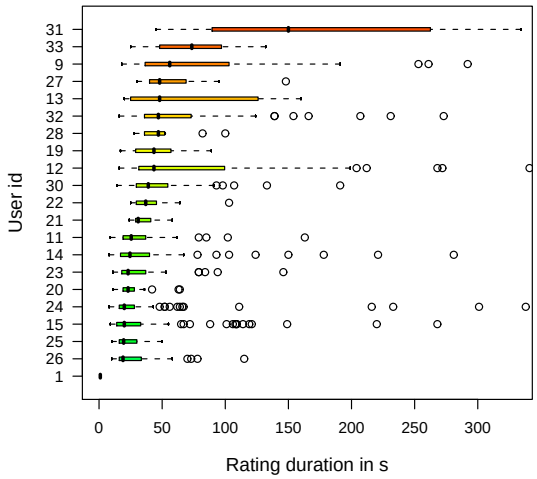


Figure 4.2: Box-plots of the rating duration for each different user (sorted by median). The average time for a rating is about one minute. Some users spend considerably more time for the task.

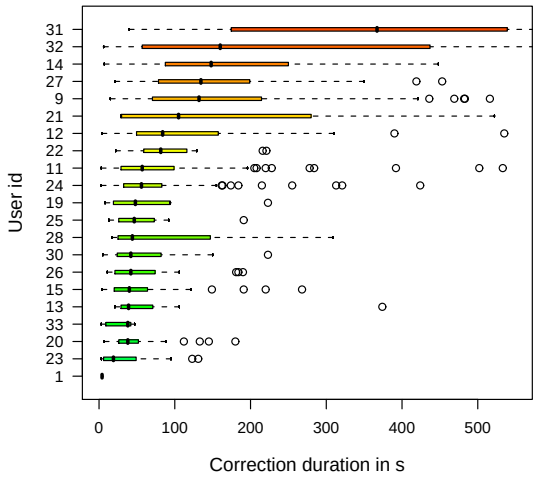
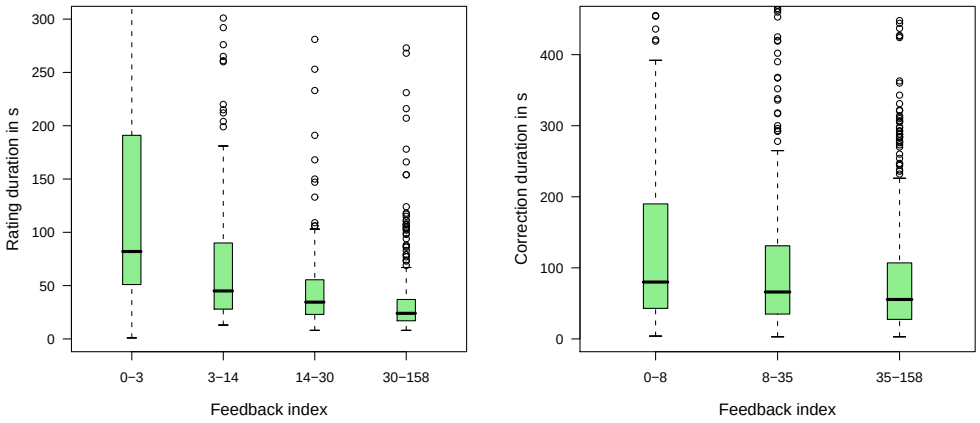


Figure 4.3: Box-plots of the correction duration for each different user (sorted by median). The average time for a rating is little more than 2 minutes. Also this figure shows big differences in user behavior.

One purpose of the feedback durations is to validate the rating scale. The time it takes a new user to give a rating depends on the quality of the translation and how easy the rating scale can be understood. New users were expected to understand the scale and adjust to it after rating a small amount of sentences. A summary of the time it took users to rate or correct sentences is shown in Figure 4.4a and 4.4b, respectively.



(a) Rating Duration (b) Correction Durations

Figure 4.4: Box-plots comparing the users' first feedback durations with later feedback duration to see the learning effect. (a) The rating duration clearly shows this effect while for (b) giving corrections a change is less visible.

The figure shows box-plots of bins of rating durations. Here, the first bin contains the duration for the first three ratings of each user. The distribution shows higher values as for the later ratings where the time settles between around 20-40 seconds. Notice that the bins are not equal in size. One problem here could be that for the higher indexes, a much lower amount of ratings were available (only half of the users rated more than 25 sentences).

Something that can be learned from such a result would be how many ratings it takes to learning the rating scale. This could be used in further experiments so that every new user has to go through training, with at least that amount of sentences.

4.1.3 Translation Quality Measures

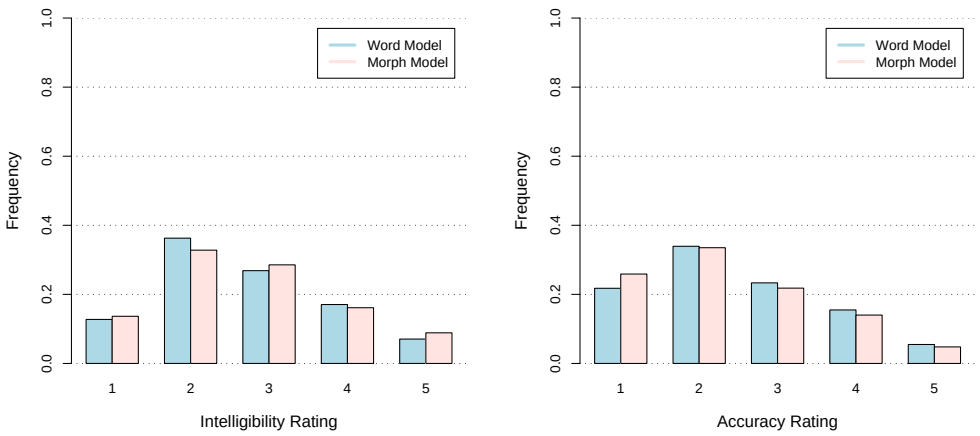
The results of the intelligibility and accuracy ratings are shown in Table 4.5. We applied the Wilcoxon rank sum test, to test statistical significance of differences between word and morph models. The null hypothesis H_0 states that no difference exist between word and morph ratings. For intelligibility, the alternative hypothesis H_1 was that the morph model got higher ratings than the word model. There was not enough evidence to reject H_0, given the one-sided test with 90% significance level (p-value 0.28). For accuracy, H_1 stated that the word model got higher ratings than the morph model. Here, H_0 can be rejected, given a 90% significance level (p-value 0.056), which means that we can state that the word model got better accuracy ratings than the morph model.

Table 4.5: Intelligibility and accuracy statistics for the human evaluation of word and morph models. Given are the median Q_2 and mean $\bar{x}$.

translation model		Q_2	$\bar{x}$
intelligibility	word	3	2.69
intelligibility	morph	3	2.74
accuracy	word	2	2.49
accuracy	morph	2	2.38

The statistical test showed that there is little evidence for differences in the word and morph model ratings. This is confirmed by the histograms in Figures 4.5a and 4.5b, which compare word and morph rating for intelligibility and accuracy, and show no big differences. Most reported intelligibility ratings are between 2 and 3. Accuracy shows more bias to the lower end with most ratings as 2. Accuracy is significantly worse than intelligibility (p-value < 0.001). The result indicates a relatively low quality of translations.

The scatter plot in Figure 4.6 suggests that good intelligibility is accompanied by good accuracy. Pearson's correlation coefficient confirmed high correlation between intelligibility and accuracy (0.7). A small number of outliers can be observed. Some are sentences that are not understandable, but convey the meaning well. This happened with very short sentences that make no sense (sentence fragments), or with sentences containing many untranslated words. On the other extreme are sentences that are well formed, but do not convey the meaning of the source text. This happened when the meaning was negated or when a translation contained nice phrases, but only few correct words, which is more typical for a PBSMT system.



(a) (b)

Figure 4.5: Histograms of (a) intelligibility and (b) accuracy distributions comparing word and morph models.

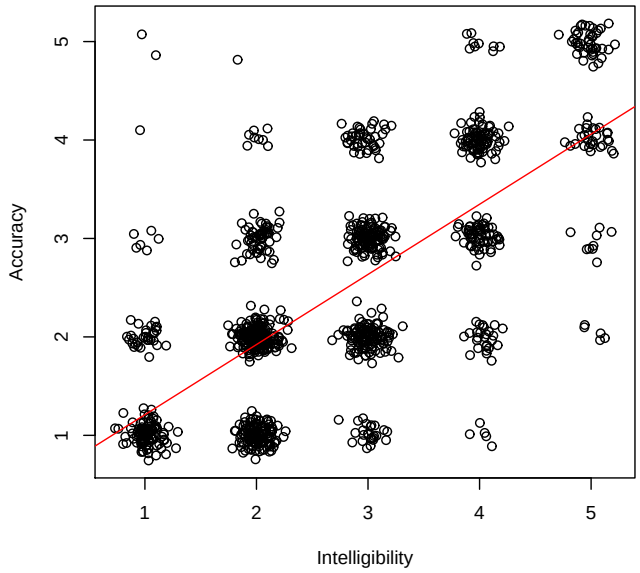


Figure 4.6: Scatter plot showing intelligibility versus accuracy. The red line shows the best fit linear regression.

4.2 Evaluation of Adaptation Models

This section shows the results achieved with the different adaptation experiments. Three different adaptation families were chosen as target of this work: *Concatenate Models*, *Interpolate Models* and *Post-edit Models* as explained in detail in Section 3.3. Section 3.3 also explains the data descriptors used for the experiments. The evaluation process including an explanation of all reported statistics is explained in Section 3.5. Some additional experiments were made using only language model adaptation. In the following, we report the baseline results, the results of the language model adaptation experiments and then show the results of each adaptation family. The section is concluded by a comparison of each family's best model.

4.2.1 Baseline Model

We compare three different baseline systems which differ in how the language model was created. The first one, $B1$, uses a language model that was created from the not pre-processed and not shortened Europarl corpus. The second, $B2$, uses a language model that created from a pre-processed corpus. The third one, $B3$, uses a language model that was created from a pre-processed and in addition had sentences longer than 100 words removed. Pre-processing is used to make the text the same as used for the translation model training.

Table 4.6 shows the BLEU scores for the different experiments. The performance of all three models is in the same range. The results of the system comparison can be found in Table 4.7. Student's t-test and Wilcoxon signed-rank test show the ranking: $B1 > B3$, whereas the bootstrap method shows no significant difference. We will choose $B2$ as our baseline for the further experiments, as the performance is not significantly different from the other two and as we can train it ourselves. For the later experiments, we need to re-train the language model (for the interpolation experiments, the Europarl and ilta-train data will be added).

The BLEU scores of the bootstrap resampling sets are summarized as histogram in Figure 4.7.

Table 4.6: Evaluation of the Iltalehti corpus test set for the baseline model using 10-fold cross-validation

Id	**Data**			**Training**	**Testing**			
	TM	RM	LM		cv	bootstrap resampling		
					mean	mean	interval	RSD%
B1	ep	ep	ep	16.61	16.55	16.55	[11.86, 22.17]	16.35
B2	ep	ep	ep	16.49	16.43	16.43	[11.69, 21.86]	16.30
B3	ep	ep	ep	16.28	16.21	16.21	[11.33, 21.69]	16.90

Table 4.7: Baseline system ranking. Is the difference in BLEU scores statistically significant? Three different statistical tests were used, bootstrap method, Wilcoxon signed-rank test and Student's t-test. For a confidence level of 95%, the latter two show a ranking: $B1 > B3$, whereas for the first one, there is not enough evidence to reject the null hypothesis and state a significant difference between the models.

comparison result	p-value
B1 - B2	0.39
B1 - B3	0.33
B2 - B3	0.48

(a) Bootstrap method

comparison result	p-value
B1 - B2	0.28
B1 > B3	0.019
B2 - B3	0.10

(b) Wilcoxon signed-rank test

comparison result	p-value
B1 - B2	0.17
B1 > B3	0.012
B2 - B3	0.071

(c) Student's t-test

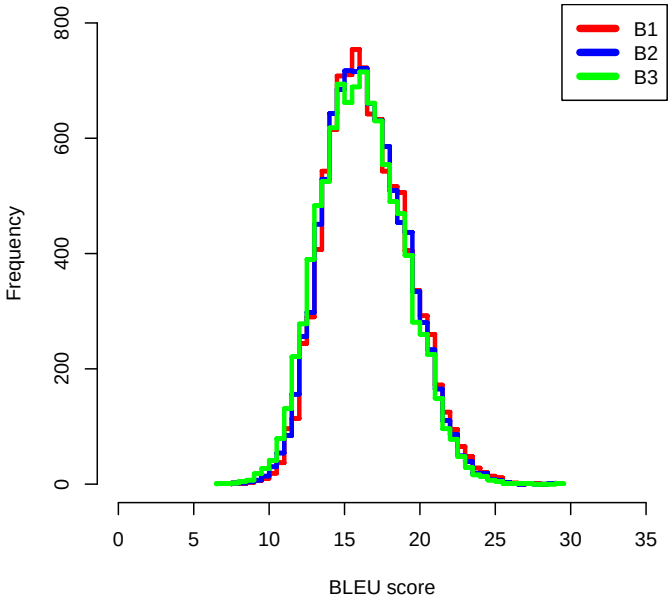


Figure 4.7: Baseline systems compared by the BLEU score histograms created from the bootstrap resampling test sets.

4.2.2 Language Model Adaptation

First we will see how an in-domain language model (LM) improves the performance. The additional data used to improve the LM is rather small with about 900 sentences. We trained the 4-gram LM, using SRILM in the same way as the one used for the baseline evaluation. Three different cases were tested. The first one, $L1$ uses the additional in-domain data by adding it to the Europarl corpus and retraining the LM with that new data.
In the next case, $L2$, an additional small 4-gram LM is created using the Iltalehti training data. During decoding, Moses combines this LM with the Europarl LM by log-linear interpolation. In the last case $L3$, the small Iltalehti training data LM is solely used. The BLEU evaluation results of these three cases are shown in Table 4.8. Table 4.9 shows the system rankings. Only considering the language model adaptation systems, the clear ranking is $L1 > L2 > L3$. However, none of these systems outperforms the baseline $B2$ in more than 95% of the samples, except that $L3$ is significantly worse than the baseline in that respect.

The BLEU scores of the bootstrap resampling sets are summarized as histogram in Figure 4.8.

Table 4.8: Evaluation of the Iltalehti corpus test set for the language model adaptation systems using 10-fold cross-validation

Id	**Data**			**Training**	**Testing**			
	TM	RM	LM		cv	bootstrap resampling		
					mean	mean	interval	RSD%
B2	ep	ep	ep	16.49	16.43	16.43	[11.69, 21.86]	16.30
L1	ep	ep	ep+il	20.50	17.25	17.25	[11.95, 22.85]	16.54
L2	ep	ep	ep, il	20.92	13.28	13.25	[8.02, 18.53]	19.44
L3	ep	ep	il	20.29	10.77	10.72	[5.96, 15.51]	21.90

Table 4.9: Language model adaptation system ranking using the same type of data as in Table 4.7. No clear ranking exists. Without considering the baseline model separately, the more pessimistic bootstrap method ranks the system as $L1 > L2 > L3$ and $B2 > L3$.

comparison result	p-value
B2 - L1	0.90
B2 - L2	0.064
B2 > L3	0.0026
L1 > L2	0.011
L1 > L3	< 0.001
L2 > L3	0.0021

(a) Bootstrap method

comparison result	p-value
B2 < L1	0.0020
B2 > L2	< 0.001
B2 > L3	< 0.001
L1 > L2	< 0.001
L1 > L3	< 0.001
L2 > L3	< 0.001

(b) Wilcoxon signed-rank test

comparison result	p-value
B2 < L1	< 0.001
B2 > L2	< 0.001
B2 > L3	< 0.001
L1 > L2	< 0.001
L1 > L3	< 0.001
L2 > L3	< 0.001

(c) Student's t-test

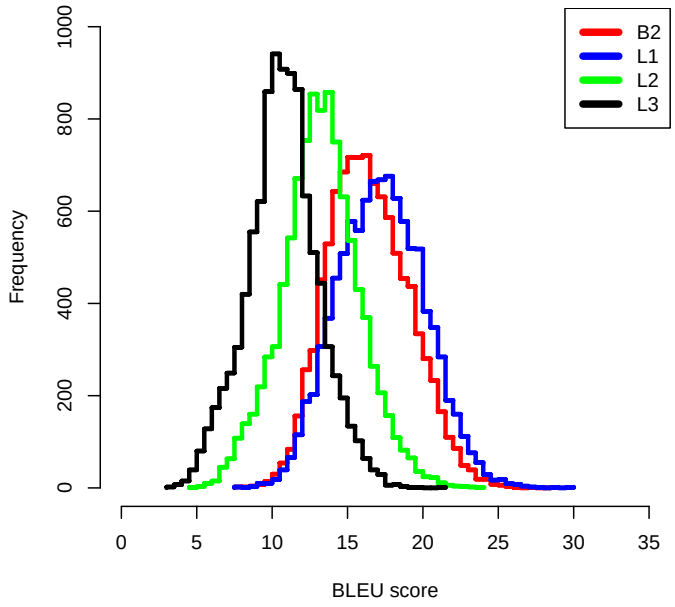


Figure 4.8: Language model adaptation systems compared by the BLEU score histograms created from the bootstrap resampling test sets.

4.2.3 Concatenate Model

In the concatenate model family of experiments, the additional Iltalehti training data is added to the existing Europarl training data. We performed two different experiments. The first one ($C1$) consisted of training the phrase and reordering tables and using the baseline Europarl LM. In the second one ($C2$), additional Iltalehti data was used to enlarge the LM. The results of the evaluation are shown in Table 4.10. The bootstrap method produces the ranking: $(C1, C2) > B2$, so the concatenate results are significantly better than the baseline.

The BLEU scores of the bootstrap resampling sets are summarized as histogram in Figure 4.9.

Table 4.10: Evaluation of the Iltalehti corpus test set for the concatenate model adaptation systems using 10-fold cross-validation

Id	**Data**			**Training**	**Testing**			
	TM	RM	LM		cv	bootstrap resampling		
					mean	mean	interval	RSD%
B2	ep	ep	ep	16.49	16.43	16.43	[11.69, 21.86]	16.30
C1	ep+il	ep+il	ep	48.92	21.41	21.41	[15.37, 28.34]	16.00
C2	ep+il	ep+il	ep+il	55.70	22.41	22.41	[15.93, 29.37]	15.90

Table 4.11: Concatenate model adaptation system ranking using the same type of data as in Table 4.7. The more pessimistic bootstrap method ranks the system as $(C1, C2) > B2$.

comparison result	p-value
B2 < C1	0.0017
B2 < C2	< 0.001
C1 - C2	0.90

(a) Bootstrap method

comparison result	p-value
B2 < C1	< 0.001
B2 < C2	< 0.001
C1 < C2	0.0029

(b) Wilcoxon signed-rank test

comparison result	p-value
B2 < C1	< 0.001
B2 < C2	< 0.001
C1 < C2	< 0.001

(c) Student's t-test

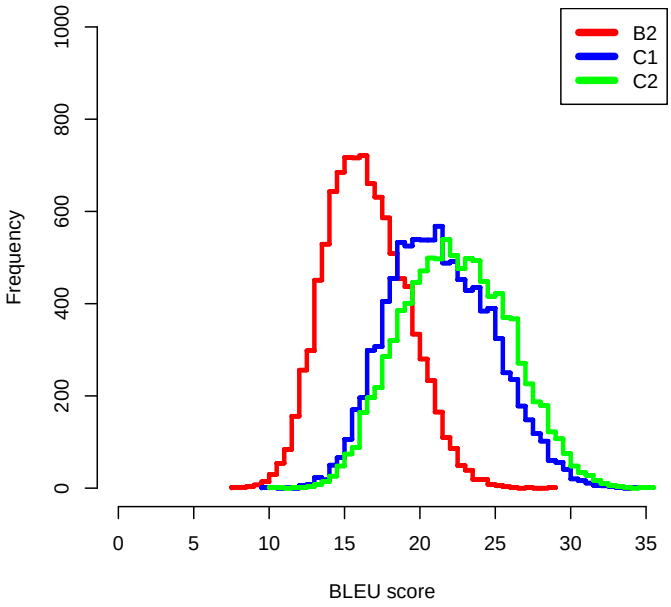


Figure 4.9: Concatenate model adaptation systems compared by the BLEU score histograms created from the bootstrap resampling test sets.

4.2.4 Interpolate Model

In the experiments of the interpolate model family, we used the baseline translation and reordering tables and the additional Iltalehti translation table. The decoder then interpolated between the baseline out-of-domain (Europarl) translation table and the adapted in-domain (Iltalehti) translation table. The table weights were set to 0.75 for the baseline system and 0.25 for the in-domain phrase table.
Similar to the concatenate models, two different experiments were conducted: one just training the phrase and reordering tables and using the baseline Europarl LM ($I1$) and another one where the LM was enlarged with the Iltalehti data ($I2$).

An additional experiment included the use of the Europarl base dictionary for the extraction of word level translation tables ($I3$). The hope here is to compensate for the small amount of training data and improve word alignment.

The results of the evaluation are shown in Table 4.12, also including the results of combining both, the Iltalehti language model and the dictionary ($I4$). The bootstrap method shows that all interpolated models are significantly better than the baseline (Table 4.13), giving the ranking: $(I1, I2, I3, I4) > B2$. However, with this method, no significant difference is found between the different versions of the interpolate models.

The BLEU scores of the bootstrap resampling sets are summarized as histogram in Figure 4.10.

Table 4.12: Evaluation of the Iltalehti corpus test set for the interpolate model adaptation systems using 10-fold cross-validation

Id	**Data**			**Training**	**Testing**			
	TM	RM	LM		cv	bootstrap resampling		
					mean	mean	interval	RSD%
B2	ep	ep	ep	16.49	16.43	16.43	[11.69, 21.86]	16.30
I1	ep, il	ep	ep	62.92	23.75	23.72	[16.79, 30.88]	15.34
I2	ep, il	ep	ep+il	68.98	24.76	24.74	[17.06, 32.75]	16.59
I3	ep, il+d	ep	ep	40.89	21.24	21.25	[15.70, 27.96]	14.82
I4	ep, il+d	ep	ep+il	43.49	21.30	21.31	[15.51, 27.87]	14.92

Table 4.13: Interpolate model mdaptation system ranking using the same type of data as in Table 4.7. The more pessimistic bootstrap method ranks the system as $(I1, I2, I3, I4) > B2$.

comparison	result		p-value
B2	<	I1	< 0.001
B2	<	I2	< 0.001
B2	<	I3	0.0032
B2	<	I4	0.0028
I1	-	I2	0.81
I1	-	I3	0.13
I1	-	I4	0.13
I2	-	I3	0.089
I2	-	I4	0.078
I3	-	I4	0.57

(a) Bootstrap method

comparison	result		p-value
B2	<	I1	< 0.001
B2	<	I2	< 0.001
B2	<	I3	< 0.001
B2	<	I4	< 0.001
I1	<	I2	0.0049
I1	>	I3	< 0.001
I1	>	I4	< 0.001
I2	>	I3	0.0029
I2	>	I4	< 0.001
I3	-	I4	0.75

(b) Wilcoxon signed-rank test

comparison	result		p-value
B2	<	I1	< 0.001
B2	<	I2	< 0.001
B2	<	I3	< 0.001
B2	<	I4	< 0.001
I1	<	I2	0.0044
I1	>	I3	< 0.001
I1	>	I4	< 0.001
I2	>	I3	< 0.001
I2	>	I4	< 0.001
I3	-	I4	0.76

(c) Student's t-test

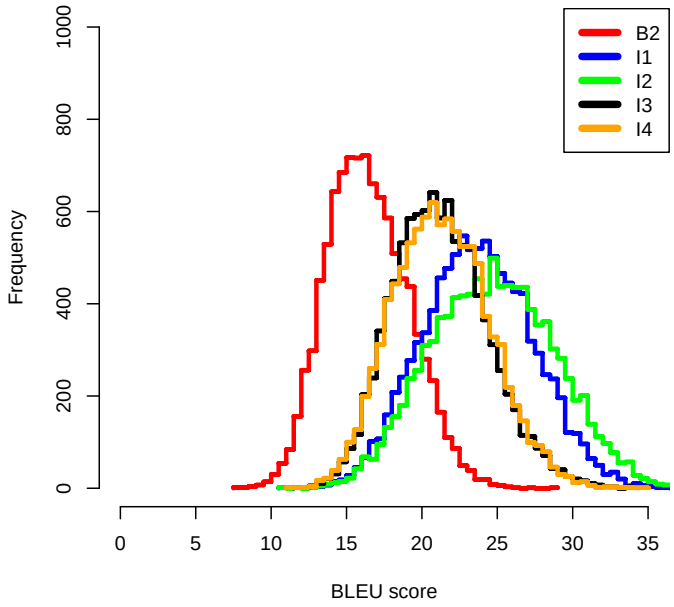


Figure 4.10: Interpolate model adaptation systems compared by the BLEU score histograms created from the bootstrap resampling test sets.

4.2.5 Post-edit Model

The experiments of the post-edit model family are different from the previous ones in that they require a two step translation. First, Finnish text is translated to "baseline English", which is then translated to English (more proper English, hopefully).

The most basic experiment uses adapted translation tables and the Europarl LM ($P1$). Then another experiment was conducted with a LM where the Iltalehti data was concatenated to the Europarl data ($P2$).

The results of the evaluation are shown in Table 4.14 Looking at the ranking Table 4.15, the testing methods are not in agreement. The bootstrap method ranks $P2 > (B2, P1)$, while the other two methods also suggest in addition that $P1 > B2$. The bootstrap method shows that $P1$ performed better than $B2$ for 94% of the samples, although the difference in BLEU score is over 7.

The BLEU scores of the bootstrap resampling sets are summarized as histogram in Figure 4.11. This system ranking is clearly reflected in the histogram.

Table 4.14: Evaluation of the Iltalehti corpus test set for the post-edit model adaptation systems using 10-fold cross-validation

Id	**Data**			**Training**	**Testing**			
	TM	RM	LM		cv	bootstrap resampling		
					mean	mean	interval	RSD%
B2	ep	ep	ep	16.49	16.43	16.43	[11.69, 21.86]	16.30
P1	pec	pec	ep	57.75	22.74	22.73	[16.71, 30.22]	15.01
P2	pec	pec	ep+il	61.02	24.05	24.04	[17.68, 32.34]	15.42

Table 4.15: Post-edit model adaptation system ranking using the same type of data as in Table 4.7. The bootstrap method gives the ranking: $P2 > (B2, P1)$.

comparison	result	p-value
B2	- P1	0.94
B2	< P2	0.035
P1	< P2	0.023

(a) Bootstrap method

comparison	result	p-value
B2	< P1	< 0.001
B2	< P2	< 0.001
P1	< P2	< 0.001

(b) Wilcoxon signed-rank test

comparison	result	p-value
B2	< P1	< 0.001
B2	< P2	< 0.001
P1	< P2	< 0.001

(c) Student's t-test

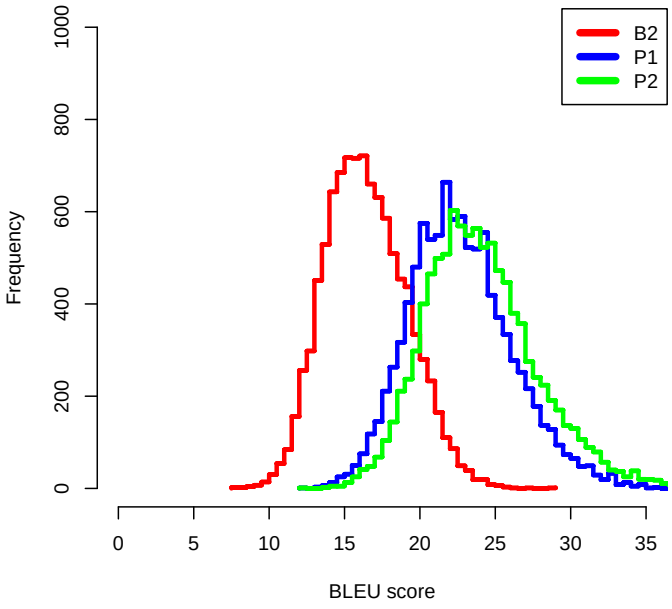


Figure 4.11: Post-edit model adaptation systems compared by the BLEU score histograms created from the bootstrap resampling test sets.

4.2.6 Model Comparison

Now let us give an overview of all conducted experiments. The results can be seen in Table 4.16. For better comparison, we only took the best versions of the different model families. The results of that model set are shown in Table 4.17. The system ranking of the best versions of each family is shown in Table 4.18. The bootstrap method does not give a clear ranking. If we consider $P2$ separately, we get the ranking $(I2, C2) > (L1, B2)$. The post-edit model only ranks better than the baseline, $P2 > B2$. Among the advanced models $(P2, I2, C2)$, no one outperforms the other in more than 95% of the samples, although there are BLEU score differences of up to 2 points. As Table 4.18 shows, the other two test methods give a more sensitive ranking.

Table 4.16: Evaluation of the Iltalehti corpus test set for all models adaptation systems using 10-fold cross-validation

Id	**Data**			**Training**	**Testing**			
	TM	RM	LM		cv	bootstrap resampling		
					mean	mean	interval	RSD%
B1	ep	ep	ep	16.61	16.55	16.55	[11.86, 22.17]	16.35
B2	ep	ep	ep	16.49	16.43	16.43	[11.69, 21.86]	16.30
B3	ep	ep	ep	16.28	16.21	16.21	[11.33, 21.69]	16.90
L1	ep	ep	ep+il	20.50	17.25	17.25	[11.95, 22.85]	16.54
L2	ep	ep	ep, il	20.92	13.28	13.25	[8.02, 18.53]	19.44
L3	ep	ep	il	20.29	10.77	10.72	[5.96, 15.51]	21.90
C1	ep+il	ep+il	ep	48.92	21.41	21.41	[15.37, 28.34]	16.00
C2	ep+il	ep+il	ep+il	55.70	22.41	22.41	[15.93, 29.37]	15.90
I1	ep, il	ep	ep	62.92	23.75	23.72	[16.79, 30.88]	15.34
I2	ep, il	ep	ep+il	68.98	24.76	24.74	[17.06, 32.75]	16.59
I3	ep, il+d	ep	ep	40.89	21.24	21.25	[15.70, 27.96]	14.82
I4	ep, il+d	ep	ep+il	43.49	21.30	21.31	[15.51, 27.87]	14.92
P1	pec	pec	ep	57.75	22.74	22.73	[16.71, 30.22]	15.01
P2	pec	pec	ep+il	61.02	24.05	24.04	[17.68, 32.34]	15.42

Table 4.17: Evaluation of the Iltalehti corpus test set for the best models of each family.

Id	**Data**			**Training**	**Testing**			
	TM	RM	LM		cv	bootstrap resampling		
					mean	mean	interval	RSD%
B2	ep	ep	ep	16.49	16.43	16.43	[11.69, 21.86]	16.30
L1	ep	ep	ep+il	20.50	17.25	17.25	[11.95, 22.85]	16.54
C2	ep+il	ep+il	ep+il	55.70	22.41	22.41	[15.93, 29.37]	15.90
I2	ep, il	ep	ep+il	68.98	24.76	24.74	[17.06, 32.75]	16.59
P2	pec	pec	ep+il	61.02	24.05	24.04	[17.68, 32.34]	15.42

Table 4.18: A system ranking of the best systems of each family using the same type of data as in Table 4.7. Using the more pessimistic bootstrap method and considering $P2$ separately, we get the ranking $(I2, C2) > (L1, B2)$. For $P2$, only the ranking $P2 > B2$ can be stated with a significance level of 95%.

comparison result			p-value
B2	-	L1	0.90
B2	<	C2	< 0.001
B2	<	I2	< 0.001
B2	<	P2	0.035
L1	<	C2	0.0013
L1	<	I2	< 0.001
L1	-	P2	0.94
C2	-	I2	0.82
C2	-	P2	0.60
I2	-	P2	0.43

(a) Bootstrap method

comparison result			p-value
B2	<	L1	0.0020
B2	<	C2	< 0.001
B2	<	I2	< 0.001
B2	<	P2	< 0.001
L1	<	C2	< 0.001
L1	<	I2	< 0.001
L1	<	P2	< 0.001
C2	<	I2	0.0029
C2	-	P2	0.90
I2	-	P2	0.35

(b) Wilcoxon signed-rank test

comparison result			p-value
B2	<	L1	< 0.001
B2	<	C2	< 0.001
B2	<	I2	< 0.001
B2	<	P2	< 0.001
L1	<	C2	< 0.001
L1	<	I2	< 0.001
L1	<	P2	< 0.001
C2	<	I2	0.0013
C2	-	P2	0.89
I2	-	P2	0.30

(c) Student's t-test

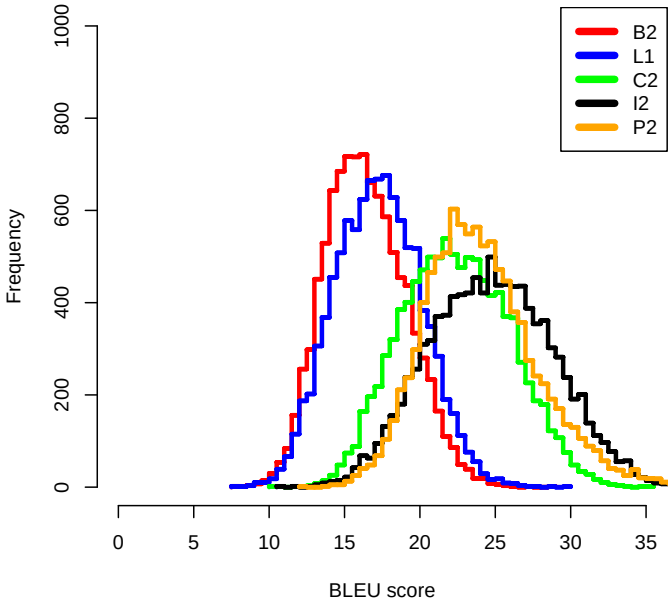


Figure 4.12: Best models of each family compared by the BLEU score histograms created from the bootstrap resampling test sets.

Chapter 5

Discussion

5.1 Adaptation Models

We expected that additional in-domain data improves in-domain translation performance. This has been confirmed for 3 of the 4 experiment families. Only the language model adaptation experiments did not score significantly higher than the baseline. Despite of this, language model adaptation has shown to boost all other experiments so that the best results within each family were achieved by using language model adaptation.

The significance intervals, relative standard deviations and histograms of concatenate and interpolate experiments show a wider range of the results than the baseline. Thus, the translation quality varies much more than before. As there is a significant improvement, some translations were much more improved than others. This might be explained by the small amount of data used for adaptation.

For the language model adaptation, we expected $L3$ (LM: il) to perform worse than the baseline, simply due to the small size of the in-domain LM. In contrast to our results, a similar experiment by Koehn and Schroeder (2007) showed a slight improvement in this case (0.77 points BLEU). However, the experiments were carried out in an easier language combination (French-English), and more importantly, the in-domain corpus had 42 000 sentence pairs, which is considerably larger than our 900 sentence pairs.

Furthermore, both language model experiments that use in- and out-of-domain data were expected to improve on the baseline results. This was not the case for $L2$, which used two separate language models as different features in the log-linear translation model and scored 3 BLEU points worse

than the baseline. The results by Koehn and Schroeder in a similar case (the same language model setup, but using all data for translation model training) were similar to a setup with linearly interpolated language model and over 2 BLEU points higher than their baseline performance. In order to compare the results, we could re-investigate this language model setup with the concatenated translation model. A possible reason for the decreased performance in this case could be a bad choice of language model feature weights (0.6 in-domain vs. 0.4 out-of-domain).

Using the concatenated language model setup (LM: ep+il) consistently increases the performance about 1 BLEU point for most experiments (except for $I4$, where we suspect a mistake in the setup). However, the improvement is not significant based on the test described in Koehn (2004). This result confirms other studies, which suggest that the language model is an effective mechanism for domain adaptation (Xu et al., 2007; Zhao et al., 2004; Koehn and Schroeder, 2007). Due to the little improvement given by the concatenated language model and the fact it is computationally expensive to create, as additional in-domain data require re-training of the whole model, we would consider a linear language model interpolation approach. Koehn and Schroeder (2007) report good results using that method.

Our concatenate models $C2$ considerably improve the baseline performance by 6 BLEU points. Related studies by Koehn and Schroeder (2007) report only 1.6 points BLEU increase for this case. A cause for this difference might be that our in-domain corpus is composed of much shorter sentences (3-12 words) than the Europarl corpus (and maybe the in-domain News-Commentary corpus used by Koehn and Schroeder). This could help the statistical word alignment and improve the word dictionaries. Another reason for the larger improvement could be their larger out-of-domain corpus (1.2 million sentence pairs), which has a stronger weight compared to the small in-domain corpus. One disadvantage of the concatenate approach is that adding in-domain data requires complete model retraining, which is computationally expensive. Furthermore, it is not easy to change domains with the concatenate approach.

The interpolate models performed better than we originally expected. Although Wu et al. (2008) reports similar results in a comparable setup with log-linear interpolation (using an in-domain dictionary instead of in-domain sentence corpus), we did not expect an 8 BLEU point increase over the baseline. Koehn and Schroeder (2007) also report good translation model interpolation results, but employ factored translation models for interpolation. The result might be explained by shorter in-domain sentences, the same

effect as described for the concatenate models.

We expected the dictionary approach for models $I3$ and $I4$ to boost word alignment and thus translation performance. Although the dictionary was an out-of-domain dictionary, it should have helped to find better word alignment. We suspect that an error in the experiment setup is responsible for this result. Reinvestigation of the setup will clarify if other causes exist.

The interpolate model scores slightly higher than the concatenate model, but not significantly (using 95% significance level with the bootstrap method). We asked ourselves if one should outperform the other. In the concatenate model, very little data was added compared to the existing corpus (800 times more sentences in our Europarl corpus), that for existing words and phrases it does not have much influence on phrase probabilities. It has, however, the power to introduce new translations. Given the large amount of data, fine grained word alignment should be possible. In the interpolate model, a separate in-domain phrase table is trained from the 900 collected sentences. With this little data, we did not expect very good word alignment. However, the separate phrase table has more weight in interpolation. Given the fixed weighting of out-of-domain and in-domain phrase tables (0.75 vs. 0.25) and the fact that no optimization was performed, we are uncertain about the potential of the method. However, it is easier to adjust the out-of-domain phrase table weight in interpolation than in concatenation.

The post-edit model performs similarly as the interpolate model and provides a significant improvement over the baseline. This result is in agreement with other post-editing studies, which show good improvements over the baseline (Isabelle et al., 2007; Simard et al., 2007a,b; Dugast et al., 2007). In comparison to the mentioned research, the size of our in-domain corpus was rather small. In that respect, our experiments can be compared with the work of De Ilarraza et al. (2008), who also obtained good results when using only a small in-domain corpus.

Given comparable translation performance of concatenate and interpolate approach, we prefer the interpolate model, based on its additional advantages. Weights can be adjusted more easily and training is computationally cheap, given a small in-domain corpus. With larger corpus sizes, a possible solution is to setup several small domain specific corpora that can be used simultaneously, similar to the setup by Xu et al. (2007). Then, the domain of the input text then governs the weights of the different domain models.

When comparing the interpolate with the post-edit approach, we prefer the the interpolate model for a statistical MT system, as it provides more flexibility. From a computational point of view, the models in both approaches can

be trained in short time, but the post-edit approach demands one additional translation step. We see more growth potential in the interpolate approach, given the fact that APE translation performance is limited by the baseline quality. However, when the APE module is used as post-processing step for a rule based MT system and for a restricted domain, the post-edit approach is a good and easy way to improve translation quality.

5.2 User Feedback Data

With 1 000 user feedback submissions, a considerable amount of corrected sentences was obtained. This was only about 50% of the planned amount, but still higher than expected for the small number of contributers. Despite this considerable result, few paired translations for word and morph models were collected. That was due to the fact that we preferred getting more translations for different source sentences instead of getting two translations for the same sentence. Multiple reference sentences would also improve automatic evaluation results, but a trade off had to be made and 1 000 translations for different source sentences were considered higher value than 500 translated source sentences, as our main focus was domain adaptation.

Virpioja et al. (2007) reported no improvement of their morph-based translation setup compared to a word-based system using BLEU evaluation. Our hope was to be able to report an improvement using human evaluation, which could not be met.

The collected data contained some unexpected tokens, which was only discovered at a later stage, where including additional filtering would have required much more additional work. Users reported several translation alternatives separated by '/' of included additional words in parentheses. Also descriptions of the nature of a translation were included in the correction text, such as "bad grammar (a very difficult sentence)". Improved instructions for the volunteers or a short feedback training stage could have avoided these cases. Improvements regarding inter-rater reliability might have been achieved by proper training of the users for the rating scales. A formal method for intelligibility and accuracy scale construction might have improved the results. A standard psychophysical method known as “equal-appearing intervals", which is suggested in Carroll (1966), could have been used.

The feedback of users indicated that some translations were too hard. Too much time was needed to find the correct English expressions. One solution could be to split translations into phrases instead of whole sentences.

Chapter 6

Conclusions

We collected a 1 000 sentences bilingual Finnish-English news corpus, which is a good asset for our machine translation research and will be re-used for further adaptation experiments. Another created asset, the feedback web application, can be re-used for further experiments in human translation evaluation. These two results contribute to objective (1) of this thesis (the thesis objectives are set in Section 1.1).

Using the collected data, we significantly improved the machine translation performance for the baseline system on the news task, contributing to thesis objective (2). Our best result was achieved using concatenation of in-domain and out-of-domain data for language model creation and interpolation of separate in-domain and out-of-domain translation models in a log-linear framework. Our outcome confirms other research in this area, which is remarkable when considering the small size of our in-domain corpus.

We conclude that translation performance can be effectively adjusted to users' needs by community feedback with relatively simple means. Given the respectable corpus collection result, we conclude that our little community experiment has succeeded. We have examined what is required for motivating users in a virtual community, providing answers to thesis objective (4). Validating these findings empirically would be a separate project on its own.

Unfortunately due to time constraints, thesis objective (3), a second round of human evaluation, could not be conducted. The volunteers would get the domain adapted translation to be evaluated and the results could be compared to our automatic BLEU evaluation. This would be a next step in validating the improvement of the adaptation effort.

Subsequent experiments will investigate the reasons of the low performance

for the interpolate model using a dictionary. Then, linearly interpolated language models should be compared to the concatenate approach we used.

After collecting a little more data, a training set can be defined, which allows us to tune the models. A series of weight combinations could be tested to find the optimal range of language model and translation model weights for interpolation. This will reveal the real potential of the different families.

In order to find the best ways to improve our results, an error analysis of a subset of translations could be performed. This could categorize errors into error classes, which would then help to identify the worst problems as was done by Dugast et al. (2007).

An improved BLEU score, however, does not yet prove that adaptation would be successful from the users' point of view. It seems quite hard to improve the quality a lot when considering, that twice as much parallel data gives a performance boost of about 2.5 BLEU points (Och, 2005). Apparently, this finding does not match our experiment results, as we got about 8 BLEU points improvement with a very small corpus. It could be that a big improvement can be achieved when beginning to add in-domain data to a new domain, but that the improvement decreases with a larger in-domain corpus. In order to keep the volunteers motivation up, any changes they make should have some influence on the translation quality. Considering the large data requirements, that is rather unlikely with the used approaches. New ideas might be needed to extract more information from a human correction and maybe weigh corrections based on user reputation. Also, a more transparent translation system could help remove noise by allowing users to comment on source sentences, which contributed to a certain phrase alignment.

Another problem with domain adaptation is that it is hard to measure the domain of a text. The concept of a domain is maybe more continuous than discrete. Depending on what features we use to describe a domain, a large "general" corpus could be split into few or many sub-domains. So would it be better to create many sub-corpora for different domains out of the large, general corpus?

Having collected a Finnish-English in-domain news corpus is a decent way to evaluate our experiments. But other ways would have existed to conduct experiments on domain adaptation without explicit bilingual corpus (compare Section 2.3). Leaving the time consuming collection process away would have allowed to concentrate on improving the domain adaptation performance. However, our more widespread approach helped to clarify the original vision of a free translation community.

Bibliography

Alegria, I., de Ilarraza, A. D., Labaka, G., Lersundi, M., Mayor, A., and Sarasola, K. (2007). Transfer-based MT from Spanish into Basque: reusability, standardization and open source. In *LNCS 4394*. Cicling.

Automatic Language Processing Advisory Committee (1966). Language and machines: Computers in translation and linguistics. Publication No. 1416, National Academy of Sciences.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Bar-Hillel, Y. (1960). The present status of automatic translation of languages. *Advances in Computers*, pages 91–163.

Béchet, F., Mori, R. D., and Janiszek, D. (2004). Data augmentation and language model adaptation using singular value decomposition. *Pattern Recognition Letters*, 25(1):15–19.

Bitzer, J., Schrettl, W., and Schroder, P. J. (2007). Intrinsic motivation in open source software development. *Journal of Comparative Economics*, 35(1):160–169.

Bojar, O. (2007). English-to-Czech factored machine translation. In *ACL Workshop on Statistical Machine Translation 2007*, pages 232–239, Prague, Czech Republic.

Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. (2007). Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867.

Brown, P. F., Pietra, S. D., Pietra, V. J. D., and Mercer, R. L. (1994). The mathematic of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the role of BLEU in machine translation research. In *EACL-2006: 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy.

Carroll, J. B. (1966). An experiment in evaluating the quality of translations. *Mechanical Translation and Computational Linguistics*, 9(3, 4):55–66.

Cheung, P. and Fung, P. (2004). Sentence alignment in parallel, comparable, and quasi-comparable corpora.

Civera, J. and Juan, A. (2007). Domain adaptation in statistical machine translation with mixture modelling. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Creutz, M. and Lagus, K. (2007). Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1). Article No. 3.

Dang, H. T., Lin, J., and Kelly, D. (2007). Overview of the TREC 2006 question answering track. In Voorhees, E. M., editor, *Proceedings TREC 2006*, volume SP 500-272.

De Ilarraza, A. D., Labaka, G., and Sarasola, K. (2008). Statistical post-editing: a valuable method in domain adaptation of RBMT systems for less-resourced languages. In *Mixing Approaches to Machine Translation. MATMT2008. Proceedings.*, pages 35–40.

Django Software Foundation (2008). Django: The web framework for perfectionists with deadlines. http://www.djangoproject.com/, [Accessed 20.11.2008].

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of Human Language Technology conference (HLT-2002)*, San Diego, California.

Dugast, L., Senellart, J., and Koehn, P. (2007). Statistical post-editing on SYSTRAN's rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220–223, Prague, Czech Republic.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Efron, B. and Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1(1):54–75.

Evert, S. (2006). How random is a corpus? The library metaphor. *Zeitschrift für Anglistik und Amerikanistik*, 54(2):177–190.

Foster, G. and Kuhn, R. (2007). Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic. Association for Computational Linguistics.

Gale, W. A. and Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Gerber, L. (2001). Working toward success in machine translation. In *Proceedings of the MT Summit VIII*, Santiago de Compostela, Spain.

Giménez, J. and Amigó, E. (2006). IQMT: A framework for automatic machine translation evaluation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*.

Hildebrand, A. S., Eck, M., Vogel, S., and Waibel, A. (2005). Adaptation of the translation model for statistical machine translation based on information retrieval. In *Proceedings of the EAMT 2005*, Budapest, Hungary.

Hutchins, W. J. (1995). Machine translation: A brief history. In Koerner, E. and Asher, R., editors, *Concise history of the language sciences: from the Sumerians to the cognitivists*, pages 431–445. Pergamon Press, Oxford.

Isabelle, P., Goutte, C., and Simard, M. (2007). Domain adaptation of MT systems through automatic post-editing. In *MT Summit XI*, pages 255–261, Copenhagen, Denmark.

Iyer, R. and Ostendorf, M. (1996). Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *IEEE Transactions on Speech and Audio Processing*, pages 236–239.

Johnson, H., Martin, J., Foster, G., and Kuhn, R. (2007). Improving translation quality by discarding most of the phrasetable. In *Proceedings of the*

2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 967–975.

Kilgarriff, A. (2001). Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):97–133.

Kilgarriff, A. (2005). Language is never, ever, ever, random. *Corpus Linguistics and Lingustic Theory*, 1(2):263–276.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *EMNLP 2004*, Barcelona, Spain.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*, Puhket, Thailand.

Koehn, P. (2007). Evaluating evaluation lessons from the WMT 2007 shared task. In *Automatic Procedures in MT Evaluation (MT Summit XI)*, Copenhagen, Denmark.

Koehn, P. and Hoang, H. (2007). Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 868–876, Prague, Czech Republique. Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.

Koehn, P. and Schroeder, J. (2007). Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227, Prague, Czech Republic. Association for Computational Linguistics.

Koh, J., Kim, Y.-G., Butler, B., and Bock, G.-W. (2007). Encouraging participation in virtual communities. *Communications of the ACM*, 50(2):68–73.

LDC (2005). Linguistic data annotation specification: Assessment of fluency and adequacy in translations. Revision 1.5.

Lee, A. and Przybocki, M. (2005). NIST 2005 machine translation evaluation official results. official release of automatic evaluation scores for all submissions.

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Marcu, D. and Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 133–139, Morristown, NJ, USA. Association for Computational Linguistics.

Maucec, M. S., Brest, J., and Kacic, Z. (2006). Slovenian to English machine translation using corpora of different sizes and morpho-syntactic information. In *Language Technologies Conference: proceedings of the 9th International Multiconference Information Society IS 2006*, pages 222–225, Copenhagen, Denmark.

Mikheev, A. (2003). Text segmentation. In Mitkov, R., editor, *The Oxford Handbook of Computational Linguistics*, Oxford Handbooks in Linguistics, chapter 10, pages 201–218. Oxford University Press, Oxford.

Miller, G. A. and Beebe-Center, J. G. (1956). Some psychological methods for evaluating the quality of translations. *Mechanical Translation*, 3(3):73–80.

Moneglia, M. (2004). Measurements of spoken language variability in a multilingual corpus. Predictable aspects. In *Prococeeding of the 4th LREC Conference*, volume 4, pages 1419–1422. ELRA, Paris.

Munteanu, D. S. and Marcu, D. (2006). Extracting parallel sub-sentential fragments from non-parallel corpora. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 81–88, Morristown, NJ, USA. Association for Computational Linguistics.

MySQL AB (2008). MySQL 5 Community Server. http://www.mysql.com, [accessed 20.11.2008].

Nagao, M., ichi Tsujii, J., and ichi Nakamura, J. (1985). The japanese government project for machine translation. *Computational Linguistics*, 11(2-3):91–110.

Och, F. J. (1999). An efficient method for determining bilingual word classes. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 71–76, Bergen, Norway.

Och, F. J. (2005). Statistical machine translation: Foundations and recent advances. In *MT Summit X*, Phuket, Thailand.

Och, F. J. and Ney, H. (2001). Discriminative training and maximum entropy models for statistical machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302, Morristown, NJ, USA. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, F. J., Tillmann, C., and Ney, H. (1999). Improved alignment models for statistical machine translation. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, University of Maryland, College Park, MD, USA.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2001). BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center.

Rayson, P. E. (2003). *Matrix: A statistical method and software tool for linguistic analysis through corpus comparison*. PhD thesis, Lancaster University.

Riezler, S. and Maxwell, J. T. (2005). On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan. Association for Computational Linguistics.

Simard, M., Goutte, C., and Isabelle, P. (2007a). Statistical phrase-based post-editing. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007)*.

Simard, M., Ueffing, N., Isabelle, P., and Kuhn, R. (2007b). Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206.

Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-2002)*, pages 901–904., Denver, Colorado, USA.

Trujillo, A. (1999). *Translation Engines: Techniques for Machine Translation*. Springer-Verlag, Berlin Germany.

Turian, J. P., Shen, L., and Melamed, I. D. (2003). Evaluation of machine translation and its evaluation. In *Proceedings of the Machine Translation Summit IX*, pages 386–393, New Orleans, USA.

Ueffing, N., Haffari, G., and Sarkar, A. (2007a). Semi-supervised model adaptation for statistical machine translation. *Machine Translation*, 21(2):77–94.

Ueffing, N., Haffari, G., and Sarkar, A. (2007b). Transductive learning for statistical machine translation. In *Proceedings of ACL*.

van Slype, G. (1979). Critical study of methods for evaluating the quality of machine translation. Final report. Technical report, Bureau Marcel van Dijk [for] European Commission, Brussels.

Virpioja, S., Väyrynen, J. J., Creutz, M., and Sadeniemi, M. (2007). Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner. In *MT Summit XI*, pages 491–498, Copenhagen, Denmark.

von Ahn, L. (2006). Games with a purpose. *IEEE Computer Magazine*, 39(6):92–94.

von Ahn, L., Blum, M., Hopper, N., and Langford, J. (2003). CAPTCHA: Using hard AI problems for security. In *Advances in Cryptology – EUROCRYPT ' 2003*, volume 2656 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Germany.

von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'04*, pages 319–326, New York, NY, USA. ACM Press.

von Ahn, L., Liu, R., and Blum, M. (2006). Peekaboom: a game for locating objects in images. In Grinter, R. E., Rodden, T., Aoki, P. M., Cutrell, E., Jeffries, R., and Olson, G. M., editors, *Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*, pages 55–64. ACM.

Weaver, W. (1949). *Translation (1949)*. The Technology Press of the Massachussetts Institute of Technology/John Wiley, New York/Clapham & Hall (London).

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, (1):80–83.

Wu, H., Wang, H., and Zong, C. (2008). Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In *COLING 08*, pages 993–1000, Manchester, UK.

Xu, J., Deng, Y., Gao, Y., and Ney, H. (2007). Domain dependent statistical machine translation. In *Proceedings of the Machine Translation Summit XI*, pages 515–520, Copenhagen, Denmark.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530, Morristown, NJ, USA. Association for Computational Linguistics.

Yu, J., Jiang, Z., and Chan, H. C. (2007). Knowledge contribution in problem solving virtual communities: the mediating role of individual motivations. In *SIGMIS-CPR '07: Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel doctoral consortium and research conference*, pages 144–152, New York, NY, USA. ACM.

Zhang, Y., Vogel, S., , and Waibel, A. (2004). Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *Fourth International Conference on Language Resources and Evaluation (LREC)*, pages 2051–2054.

Zhao, B., Eck, M., and Vogel, S. (2004). Language model adaptation for statistical machine translation with structured query models. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 411, Morristown, NJ, USA. Association for Computational Linguistics.

Appendix A

Corpus Data

A.1 Europarl Corpus Example

Table A.1: Example paragraph in the Europarl corpus. On the left side are the Finnish sentences, which are aligned with the English sentences on the right side.

istuntokauden uudelleenavaaminen	resumption of the session
julistan perjantaina joulukuun 17. päivänä keskeytetyn euroopan parlamentin istunnon avatuksi ja esitän vielä kerran vilpittömän toiveeni siitä , että teillä olisi ollut oikein mukava joululoma .	i declare resumed the session of the european parliament adjourned on friday 17 december 1999 , and i would like once again to wish you a happy new year in the hope that you enjoyed a pleasant festive period .
kuten olette varmaan saattaneet huomata , vuodenvaihteeseen 2000 povattuja suuria tietokoneongelmia ei ilmennytkään . sen sijaan todella kauheat luonnonkatastrofit koettelivat kansalaisia joissakin unionimme maissa .	although , as you will have seen , the dreaded ' millennium bug ' failed to materialise , still the people in a number of countries suffered a series of natural disasters that truly were dreadful .
te olette esittäneet toiveen , että tästä asiasta keskusteltaisiin lähipäivinä tämän istuntojakson aikana .	you have requested a debate on this subject in the course of the next few days , during this part-session .

A.2 Differences of Europarl and Iltalehti

Table A.2: Extract of distinctive words for the Europarl out-of-domain (normative) corpus shown by the Log-likelihood ranking of word types.

	in-domain		out-of-domain		
word type	count	rel. frequency	count	rel. frequency	LL
euroopan	322	0.036	103 724	0.604	8 025.12
arvoisa	2	0.000	69 875	0.407	7 004.96
että	6 255	0.705	300 747	1.750	6 905.18
puhemies	19	0.002	55 947	0.326	5 408.71
komission	25	0.003	56 318	0.328	5 387.78
parlamentin	53	0.006	43 109	0.251	3 845.56
)	0	0.000	33 824	0.197	3 407.21
(	0	0.000	33 680	0.196	3 392.71
komissio	12	0.001	34 765	0.202	3 359.01
meidän	192	0.022	45 892	0.267	3 292.25
"	0	0.000	32 053	0.187	3 228.81
unionin	38	0.004	35 569	0.207	3 216.01
jäsen	55	0.006	30 255	0.176	2 574.97
jotka	729	0.082	54 875	0.319	2 153.57
neuvoston	13	0.001	22 564	0.131	2 131.36
haluaisin	33	0.004	24 015	0.140	2 117.03
tämä	834	0.094	56 539	0.329	2 008.61
tämän	556	0.063	43 344	0.252	1 754.06
parlamentti	16	0.002	18 734	0.109	1 725.46
olemme	254	0.029	31 237	0.182	1 722.90
ja	20 886	2.353	534 646	3.111	1 712.48
yhteisön	19	0.002	17 310	0.101	1 561.23
jäsenvaltioiden	1	0.000	14 445	0.084	1 439.97
jotta	139	0.016	22 484	0.131	1 409.86
huomioon	51	0.006	15 925	0.093	1 223.53
tätä	233	0.026	24 113	0.140	1 203.04
tärkeää	64	0.007	14 160	0.082	992.72
tässä	513	0.058	31 245	0.182	988.71
me	320	0.036	24 375	0.142	966.72
mietintö	3	0.000	9 796	0.057	950.32
osalta	84	0.009	14 647	0.085	946.15
koskevan	8	0.001	10 091	0.059	934.47
sen	1 947	0.219	70 810	0.412	926.56
herra	40	0.005	12 045	0.070	917.68
unioni	6	0.001	9 352	0.054	878.00
voimme	41	0.005	11 672	0.068	877.31
emme	202	0.023	18 783	0.109	872.26

Table A.3: Extract of distinctive words for the Iltalehti in-domain corpus shown by the Log-likelihood ranking of word types.

	in-domain		out-of-domain		
word type	count	rel. frequency	count	rel. frequency	LL
nolla	1 251	0.141	40	0.000	7 187.12
markkaa	1 005	0.113	36	0.000	5 747.79
hän	3 869	0.436	16 552	0.096	5 159.41
suomen	1 172	0.132	696	0.004	4 666.80
sanoo	1 258	0.142	1 062	0.006	4 489.32
mies	674	0.076	267	0.002	2 966.57
mika	443	0.050	1	0.000	2 655.87
suomessa	589	0.066	353	0.002	2 339.33
mm	407	0.046	27	0.000	2 253.46
helsingin	621	0.070	516	0.003	2 228.24
kari	283	0.032	0	0.000	1 705.64
tv	316	0.036	30	0.000	1 703.52
poliisi	415	0.047	228	0.001	1 687.96
markan	297	0.033	26	0.000	1 611.77
tuli	630	0.071	1 146	0.007	1 602.50
sai	667	0.075	1 369	0.008	1 582.50
juha	253	0.029	0	0.000	1 524.83
pekka	260	0.029	5	0.000	1 517.92
jari	238	0.027	0	0.000	1 434.43
miehen	343	0.039	173	0.001	1 426.43
suomi	409	0.046	373	0.002	1 420.19
matti	225	0.025	3	0.000	1 324.43
noin	854	0.096	3 279	0.019	1 266.20
kello	338	0.038	271	0.002	1 227.56
kertoi	379	0.043	415	0.002	1 226.95
jukka	200	0.023	0	0.000	1 205.40
lauantaina	224	0.025	91	0.001	980.49
paavo	162	0.018	1	0.000	964.29
suomalainen	186	0.021	27	0.000	961.78
mä	159	0.018	0	0.000	958.29
elokuva	179	0.020	25	0.000	929.59
mutta	4 199	0.473	47 969	0.279	929.11
antti	154	0.017	0	0.000	928.16
eun	154	0.017	0	0.000	928.16
janne	148	0.017	0	0.000	892.00
ollut	1 826	0.206	15 622	0.091	882.18
metrin	197	0.022	79	0.000	864.77
mikko	143	0.016	0	0.000	861.86
isä	181	0.020	49	0.000	857.56

Appendix B

User Feedback Application

B.1 Invitation Letter

```
Date: Thu, 24 Jan 2008 18:27:14 +0200 (EET)
From: Marcus Dobrinkat <mdobrink@cis.hut.fi>
To: labra@james.hut.fi
Subject: Help required for statistical machine translation research

Hello All,

Please have a look at

                        http://cog.hut.fi/mtreview

You will find the machine translation review application for our
Finnish-English statistical machine translation system.

Rate some already translated English sentences and provide some correct
Finnish translations.

Your input will
        - help to gather reference translations for a news domain corpus
        - improve statistical machine translation systems by user
feedback
        - hopefully show that humans judge our morphology-aware system
[1]
          higher than the word based one

If this is not good enough for you, just log on to laugh at some of the
funny machine translations.

Background:
```

```
This application is part of my master's thesis with the title
"User feedback for domain adaption in statistical machine translation".
The bigger idea is to have a free web based translation system for
complete sentences where the users can continuously improve the system
by
adding new translations and correcting existing ones.

Thanks for the help!

Marcus Dobrinkat

[1] S. Virpioja, J. J. Väyrynen, M. Creutz, M. Sadeniemi.
Morphology-Aware
Statistical Machine Translation Based on Morphs Induced in an
Unsupervised
Manner. In Proceedings of MT Summit XI, Copenhagen, Denmark, pp.
491-498,
2007.

--
Marcus Dobrinkat
http://www.cis.hut.fi/~mdobrink/
mdobrink@cis.hut.fi
Undergraduate Researcher, CIS/HUT           gsm: 040 833 0085
```

B.2 Screenshots

This section contains screenshots from the created web application that was used to collect user feedback on translations.

Figure B.1: Screenshot showing the log-in screen for the MT review web application.

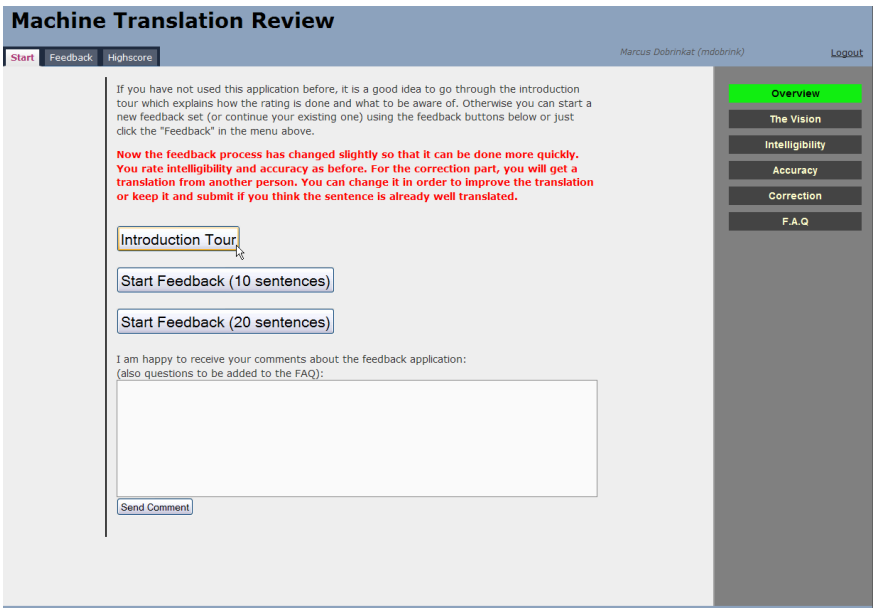


Figure B.2: Screenshot of the MT review web application showing how the translation correction was collected.

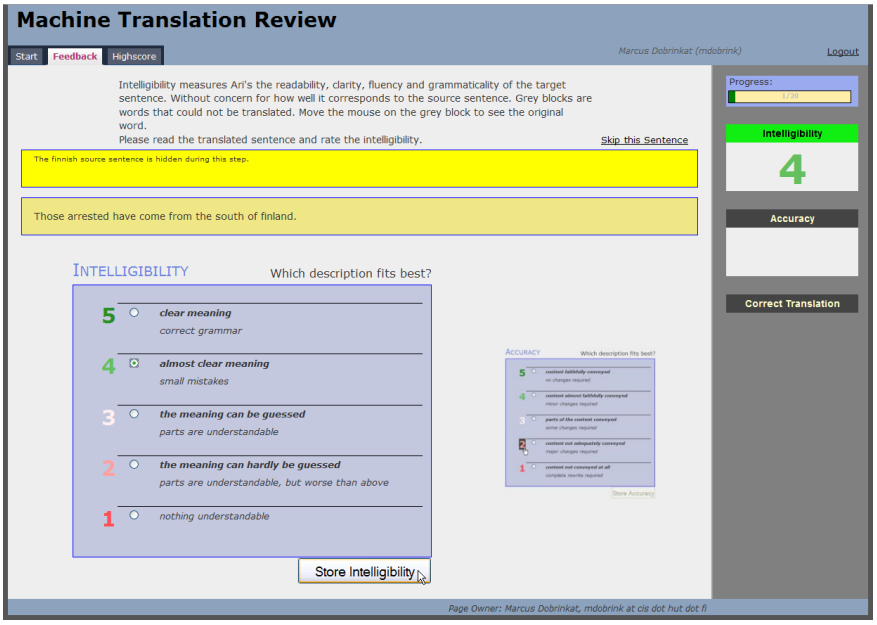


Figure B.3: Screenshot of the MT review web application showing how the intelligibility rating was collected.

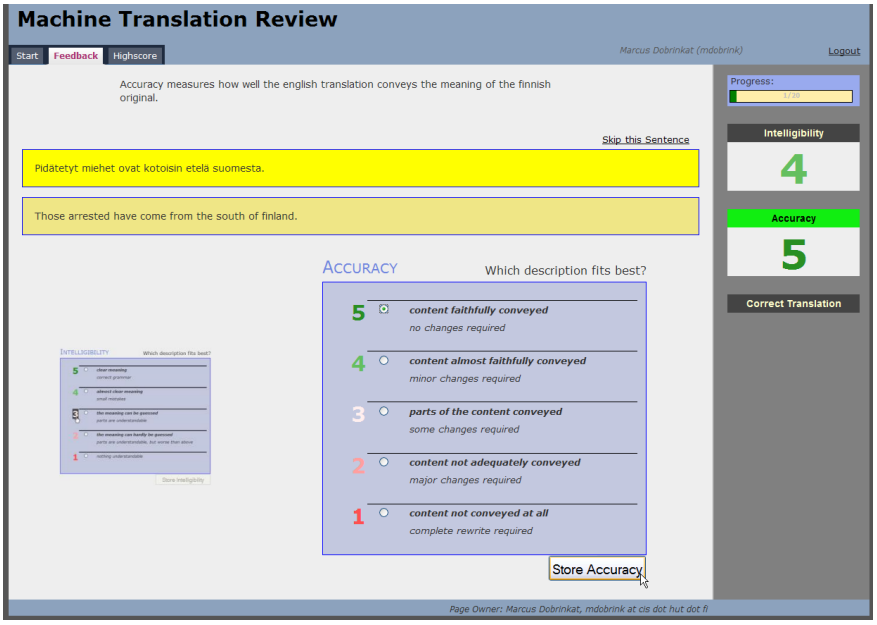


Figure B.4: Screenshot of the MT review web application showing how the accuracy rating was collected.

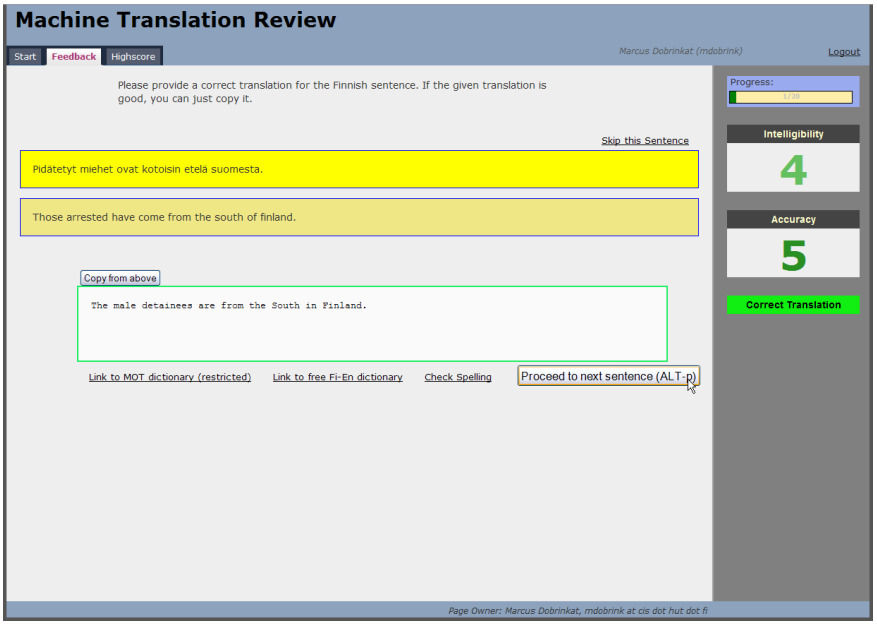


Figure B.5: Screenshot of the MT review web application showing how the translation correction was collected.

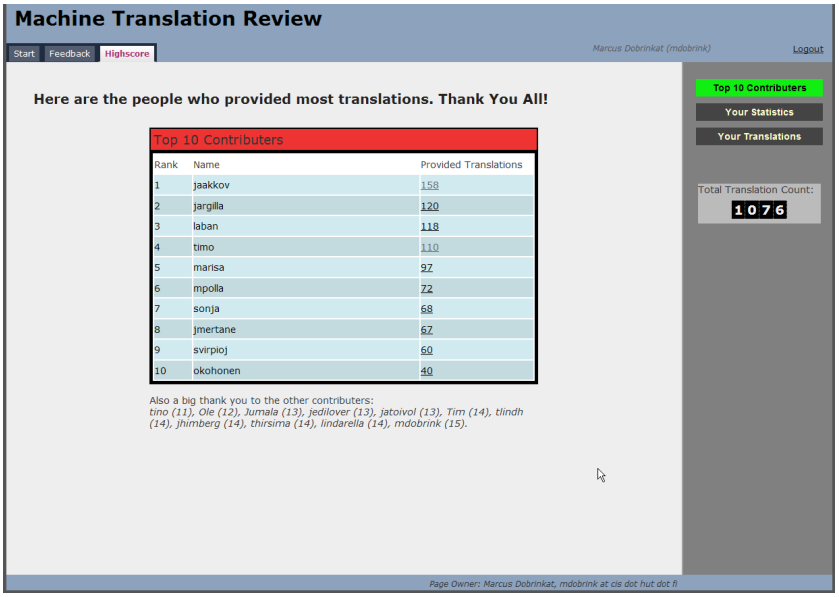


Figure B.6: Screenshot of the MT review web application showing the entry screen.

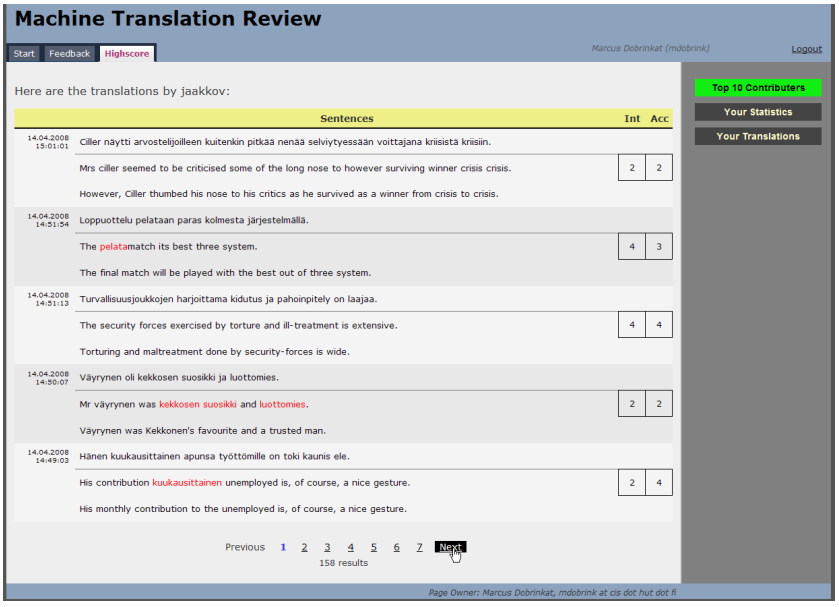


Figure B.7: Screenshot of the MT review web application showing the translations that a user entered. All entered feedback from the first 10 users in the highscore could be shown.

B.3 Data Model

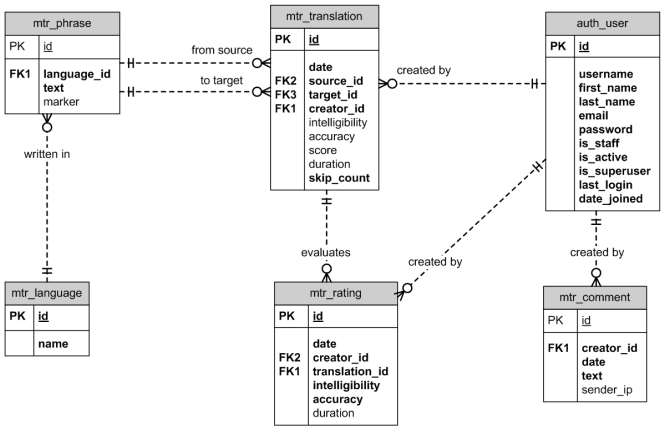


Figure B.8: User feedback application data model as entity relationship diagram.