Using EBMT to Produce Foreign Language Subtitles

Stephen Armstrong

A dissertation submitted in fulfilment of the requirements for the award of

Master of Science (M.Sc.)

to the

Dublin City University
School of Computing

Supervisor: Prof. Andy Way

January 2007

Abstract

Due to limited budgets and an ever-diminishing timeframe for the production of foreign language subtitles, pressure on subtitle companies is at an all-time high. Although translation technologies are ubiquitous in other areas of translation, especially localisation, and have been helping translators work more efficiently for a number of years now (Lagoudaki, 2006), it is strange to note that subtitle companies have been slower to jump on the bandwagon. Recent research from both academia and the industry (O'Hagan, 2003; Carroll, 2004; Gambier, 2005) suggests that the inroads made in natural language processing and machine translation could go a long way to alleviating some of this pressure.

In this thesis, we set out to establish how example-based machine translation (EBMT) can be used to speed up the subtitling process, thus improving the throughput of the subtitler, and also as a means of automatically producing foreign language subtitles which subtitle companies may not normally provide, even though they would be extremely helpful for the viewing public.

Through the development of the modular corpus-based MT engine, MaTrEx (Stroppa et al., 2006), and the collection of a large amount of subtitle data extracted from over 50 full-length features (Armstrong et al., 2006a), we were able to apply a number of EBMT techniques to produce subtitles for the language directions German–English and English–German. These machine-produced subtitles were evaluated using a range of both well-established automatic metrics common to machine translation as well as some novel manual evaluation strategies. Both automatic metrics and the human evaluation were very useful in the developmental process where we were able to isolate and fix errors made by our system. In addition, through obtaining a human's perspective on the subtitles produced by our system, we were able to gauge the acceptability of these subtitles for public viewing, and have provided a solid grounding for future research into the acceptability of (semi-) automatically generated subtitles.

Acknowledgements

Firstly, I would like to thank sincerely my supervisor, Andy Way, who was always there when I needed him. He was a great motivator at times when I was feeling lazy, and always gave constructive criticism when needed and deserved. Both Minako O'Hagan and Dorothy Kenny also deserve special thanks for all their help, advice and feedback throughout the year.

Thanks to Colm and Marian, who I could never have completed my thesis without. And to all members of the NCLT, it was a pleasure working with you all. Thanks to Bart, Karolina, Masanori, Ríona, Sara, Yvette, Yanjun, and especially Declan for never loosing the rag with all my silly questions, and Nicolas for being like my second supervisor.

Many thanks to Enterprise Ireland who generously gave me funding for the year, without which I would have wasted away on noodles.

To all the Portuguese crew who made my holiday there the best ever, Ana, Miguel, Pedro, Cuhna, David, Feenish, Céline and Sara. To Dominic, Jeff, Stew, Clément, Lucia and Daniela for being best friends. To all the organisers of ASLIB and Elina for making my few days in London very memorable.

I'd also like to say a big thank you to my favourite nieces, Taylor and Cíona, who always provided me with a fun alternative to studying. To Mark and Janice, Sharon and Rich, and Carol and Vin for all the study relief.

I've saved my BIGGEST THANKS for my Mam and Dad. You've both had the patience of saints over the years (which I know is often tested), and have always been there to give me support and encouragement.

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Master of Science is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

(Stephen Armstrong)

ID No.: 99356520

Date: 6/02/07

Contents

Chapter 1

Introduction

This research has its origins in a preliminary study by Minako O'Hagan (2003) where she noted the huge amounts of pressure subtitlers are being put under by film companies to produce high-quality subtitles in an ever-diminishing amount of time. Apparently after the Japanese cinema release of the first in the Lord of the Rings trilogy, *The Fellowship of the Ring*, the distributors and the director himself, Peter Jackson, received an overwhelming amount of criticism about the quality of the Japanese subtitles. It turned out that these subtitles were translated by just one person, who had to work on several other projects at the same time, and only had a week to do so, which was nowhere near enough time to do proper background research on the film and the type of language used.

> "One of the worst nightmares happened with one of the biggest titles for this summer season. I received five preliminary versions in the span of two weeks and the so-called 'final version' arrived hand-carried just one day before the Japan premiere." (Toda, 2005)

The conclusions of this study suggest that this scenario is all too common in the subtitling community and that some kind of automatic translation solution could go a long way to alleviating at least some of this pressure.

Translation memory tools have generally been accepted by the translation community, are ubiquitous in large-scale localisation companies, and are also used by the majority of freelance translators. Even though it is obvious that translation technology plays a major part in the translation industry today, subtitling companies have been somewhat slower in jumping on the translation technology bandwagon. The translation technology solution we propose is the use of an example-based machine translation (EBMT) system which can be used either in conjunction with a human subtitler, or indeed to fully automate the process. This leads us onto a very important point with regards to the aim of our research: we by no means intend to use our EBMT system to replace the subtitler. Instead, we envisage how EBMT could be used similar to a translation memory tool, where the subtitler could improve their throughput by post-editing the output produced by our system. It has been shown before (Joscelyne, 2006; Pigott, 1988; Wagner, 1985) that machine translation plus post-editing results in a faster turn-around time than when translating from scratch, providing the quality of the machine-produced output is of high enough quality. Secondly as EBMT can also be used without a human, it could be extremely helpful in producing foreign language subtitles in cases where they do not already exist. Consider DVD bonus material for example, which is normally only subtitled in the source language, if indeed it is subtitled at all, or for in-flight movies and film festivals which are shown to a highly diverse audience who may have no knowledge at all of the original language yet would find subtitles in their own language a very useful aid.

The language pair we chose to test our system with was English–German. DVD sales in the English- and German-speaking countries are extremely healthy with billions of dollars worth being sold each year. Furthermore, all English-speaking countries use subtitles rather than dubbing as a means of translating foreign language movies. Although foreign language films in German-speaking countries tend to contain a dubbed soundtrack, subtitling is becoming increasingly popular especially with the younger generation, and in fact foreign language DVDs imported

into the country will contain the original language subtitles along with the translated German subtitles. Even though we use DVD subtitles as our standard, the exact same techniques used to generate these subtitles could easily be applied to future types of digital media such as Blue-ray Disc and HDDVD (high-definition DVD), which share the same basic structure as a regular DVD but differ in that each are able to store significantly larger amounts of data.

Thesis structure This thesis is structured as follows. In chapter 2 we introduce the paradigm of example-based machine translation (EBMT) and explain why we believe machine translation is particularly well-suited to the task of translating subtitles. We then document the system we developed in chapter 3. As our system relies heavily on aligned bilingual data, in chapter 4 we detail the various corpora we gathered together to train and test our system. In chapter 5 we discuss the various experiments we conducted during the developmental and post-developmental stages of the project. Finally we conclude by summarizing our work to date and by mentioning some possible future avenues for future research. The following gives a more detailed description of the material we present.

Chapter 2 The notion of EBMT was first introduced by Nagao (1984). This is just one approach to machine translation, with the other two main approaches being rule-based machine translation (RBMT) and statistical machine translation (SMT). We begin this chapter by explaining how subtitles can be good candidates for MT due to constraints imposed on the subtitler as well as the repetitive nature of language across similar scenes in movies. We then go on to discuss EBMT in detail and illustrate how this works with the use of some simple examples. Finally we compare and contrast EBMT with other existing translation technology solutions such as translation memory (TM) systems, RBMT and SMT.

Chapter 3 In this chapter we give a detailed description of the corpus-based MT system, MaTrEx (Stroppa et al., 2006), that we developed and adapted to deal

with the task of translating DVD subtitles. We begin by giving an overview of the system and its components. The system is developed using a modular architecture, and we duly proceed by analysing each module in turn: word-alignment, chunking, chunk-alignment and decoding.

Chapter 4 MaTrEx is corpus-based machine translation system, which means that it needs to be trained on an aligned bilingual corpus. The nature of the language of this corpus will have a huge impact on translation quality as examples are directly drawn from this and then recombined to form an output translation. We gathered together two types of corpora to train our system: subtitle data, and data from the Europarl corpus (Koehn, 2005). By extracting source and target language subtitles from over 50 full-length features using techniques similar to optical character recognition (OCR), we were able to convert these subtitles to text format, align these at the sentence level, and produce an aligned bilingual corpus of subtitles. Firstly we detail this process and then go on to compare our subtitle data to the other corpus on which we experimented which is made up of parliamentary proceedings from the European Parliament. We perform basic analysis on both corpora, calculating the number of sentences, types, tokens and average sentence length. Finally, using the chi-square test (Kilgarriff, 2001), we measure how similar (or as in our case, dissimilar) our corpora are to each other.

Chapter 5 In this chapter we describe the various experiments and evaluation studies we conducted both during and after the developmental stage of our system. We use automatic evaluation metrics common in the area of MT as a means of determining the effect of changes made to the system on the translation quality of our system output. Such changes to the system include using different types and different amounts of corpora to train the system and also using different combinations of alignment techniques to determine what alignment strategies work best for the language pair used in our experiments. We then use the same automatic metrics to determine how well our system copes with DVD bonus material. In addition to the

automatic metrics, we also conduct two types of evaluation from which we are able to capture a human's perspective on the system output. Our aim is to highlight the types of errors made by our system and also gauge the acceptability of the subtitles produced by our system for public viewing.

Chapter 6 Finally, we conclude and give some avenues for future research.

Chapter 2

Our approach to automating subtitles: EBMT

The aim of our research is twofold. Firstly we envisage a technology-based solution which would help improve the throughput of the subtitler. We see our system as working similar to a translation memory tool, where the machine-produced output could be post-edited with the help of the subtitler where necessary. This would hopefully result in a quicker translation turnaround time, as it has been shown before (Joscelyne, 2006; Pigott, 1988; Wagner, 1985) that machine translation plus post-editing results in a faster turn-around time than when translating from scratch, with the issue of the post-editing effort being discussed in detail by O'Brien (2006). Secondly, we want a solution that will produce foreign language subtitles in situations where they do not already exist. Such cases are for certain bonus material where subtitles may only exist in the source language, if they exist at all, or for in-flight movies and film premieres where no subtitles usually exist, even though these movies are shown to an extremely diverse audience where subtitles might be helpful.

We believe that machine translation could be used to achieve these two goals and outline the suitability of MT for the translation of subtitles in section 2.1. The approach we take to MT is example-based which we describe in section 2.2, and

explain the differences between this and other translation technology approaches in section 2.3.

2.1 Suitability of MT for subtitle translation

Machine translation has been successfully applied to deal with a number of tasks, including the automatic translation of weather reports (Chandioux, 1976), medical catalogues (Muegge, 2006), patent applications (Povlsen and Bech, 2001), and technical manuals (Kamprath et al., 1998). What these tasks have in common is that they are all specific to a certain controlled-language (Way and Gough, 2005), whose grammar and lexicons have been simplified in order to ease the readability of the translation and also make it easier for the machine translation system to process the input sentences. Although movies are an art-form and the dialogue can often appear lexically rich, certain restrictions are actually placed on the subtitler where the subtitles produced can be said to follow at least a few traits of a controlled-language.

Subtitlers work under the constraints that they are only allowed 32 characters per line, and that no more than 2 lines should be shown on screen at the same time. It is generally preferred that, where possible, a sentence be spread across two lines rather than continue onto another subtitle, as this can become confusing for the viewer. In order for the subtitler to accommodate these factors, simpler syntactic forms are sometimes preferred as they result in shorter sentences, colloquial phrases dropped, and low frequency words replaced with higher frequency words with similar meaning. This ultimately makes the subtitles more quickly and easily understood. (cf. Figure 2.1 for some examples of the shortening of dialogue for subtitles). From the data we gathered to train our system described in chapter 4, we were able to calculate that our subtitle data had an average length of 11 words per sentence. This contrasts heavily with the Europarl corpus, which we also used to train the system in separate experiments, where the average length of a sentence was over 35 words.

Punctuation also differs greatly to normal language use, where the subtitler must follow a number of rules which are not necessarily the same as in normal language use. Some of these rules include the addition of 'sequence dots' (...) at the end of a line to indicate an incomplete sentence, 'italics' are used to signal foreign words that do not need to be translated, and 'n-dashes' (–) are used to note a change of speaker.

DIALOGUE VERSUS SUBTITLES

Original Dialogue	In fact he met privately with the president, though unfortunately there wasn't enough time for a photo opportunity.
Subtitle	He met with the president, but there wasn't enough time for a photo opportunity.
Original Dialogue	To tell you the truth...
Subtitle	Truthfully...
Original Dialogue	I just thought it was fair to tell you that Gilbert and I will be submitting this to the league and asking them to set aside the round.
Subtitle	I thought I should tell you that Gilbert and I will be asking the league to set aside the round.
Original Dialogue	Call me when you get home and I'll send a car for you.
Subtitle	Call and I'll send a car.
Original Dialogue	They're a bunch of fucking amateurs.
Subtitle	They're amateurs.
Original Dialogue	They posted the next round of the tournament
Subtitle	They posted the next round.

Table 2.1: Some examples taken from the movie *The Big Lebowski*, where the dialogue has been significantly shortened when it came to subtitling in order to make it more readable.

Furthermore a study by Taylor (2006) suggests that language use may appear quite predictable across certain scenes in movies. The methodology he presents is based on isolating and extrapolating specific scene types from a range of films, where similar scenes were then studied with respect to the number of co-occurring words and phrases within each scene type. For example, you might have a scene in a restaurant in one movie and a similar scene but in a different movie, where the same phrases are being used over and over. This theory of the predictability of language in particular situations is supported by Sinclair (1991) with respect to corpus linguistics, and Hoey (2006) in terms of his 'priming' hypothesis. Taylor mentions that this notion of predictability has serendipitously led him to distinguish the difference between the 'artistic film' and the 'more mundane variety', where he proposes that "the more run-of-the-mill productions could be candidates for a sophisticated kind

of translation memory tool".

Both these factors, (a) that subtitles are somewhat 'controlled', and (b) that subtitles can be predictable, imply that we should know a good deal about what kind of text is to be expected in the subtitling domain, which makes them highly amenable to machine translation, as the more linguistic knowledge we have about the source and target languages the better the translation quality should be.

2.2 EBMT - an overview

The approach we take to the automatic translation of subtitles is example-based machine translation (EBMT). The idea of EBMT dates back as far as the early 1980s, where Makato Nagao presented a paper at a 1981 conference, which was not published until 3 years later (Nagao, 1984). EBMT has appeared under several guises with the approach originally being coined by Nagao as "machine translation by example-guided inference, or machine translation by the analogy principle". The basic premise can be summed up as follows:

> "Man does not translate a simple sentence by doing deep linguistic analysis, rather, man does translation, first, by properly decomposing an input sentence into certain fragmental phrases, ... then by translating these phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference." (Nagao, 1984:178f.)

This is based on the intuition that we humans make use of previously seen translation examples to translate unseen input. EBMT is essentially split into two main processes: (1) training and (2) translation. EBMT is a corpus-based technique and during training it makes use of an aligned bilingual corpus, usually aligned at sentence level, from which we are able to extract a database of subsentential examples. This example database consists of aligned sentences, phrases and words. During translation the input sentence is matched against the source side of the example

database, where useful fragments are identified, their target language counterparts extracted, which are then recombined to produce a final target language translation.

2.2.1 Marker-based EBMT

Different methods such as phrase-structure (sub-)trees (Hearne and Way, 2003), dependency trees (Watanabe et al., 2003), and placeables[1] (Brown, 1999) have all been used in previous research to identify subsentential examples. An alternative approach, and the approach we favour (Gough and Way, 2004; Way and Gough, 2005; Groves and Way, 2005; Armstrong et al., 2006a), is to use a set of *marker* words which are used as an indicator of where one linguistic unit ends and the next one begins. This series of research papers is based on the *marker hypothesis* (Green, 1979), which is a universal psycholinguistic contraint that states that all natural languages are 'marked' for syntactic structure at surface level by a closed set of specific lexemes and morphemes, such as determiners, prepositions and pronouns. During training of the system the source-target aligned sentences are segmented at each new occurrence of a marker word, and using a number of similarity metrics a set of aligned source-target chunks is produced. Consider the English sentence in (2.1):

(2.1) I love going to the cinema

We use the marker tags ⟨DET⟩, ⟨PERS_PRO⟩ and ⟨PREP⟩ to segment the sentence in (2.1) into smaller units as in (2.2):

(2.2) ⟨PERS_PRO⟩ I love going ⟨PREP⟩ to ⟨DET⟩ the cinema

Although our sentence is now chunked into smaller units, some units are not that helpful. In order to ensure we get the most information possible from our chunks,

[1]Linguistically tagged entries are added to an example database by hand, which permits recursive matches that replace the matched text with the associated tag. For example, consider the sentence: *John Miller flew to Frankfurt.* This becomes: ⟨*firstname-m*⟩ ⟨*lastname*⟩ *flew to* ⟨*city*⟩ *on* ⟨*month*⟩ ⟨*ordinal*⟩, after an initial pass, and then: ⟨*person-m*⟩ *flew to* ⟨*city*⟩ *on* ⟨*date*⟩, after a second pass. This tokenised form will now match a sentence such as: *Dr. Howard Johnson flew to Ithaca on 7 April 1997*, along with many other possibilities.

we apply the rule that each chunk must contain at least one non-marker word. In cases where marker words appear consecutively, we keep the first marker tag, and discard the remaining tags. As the marker words 'to' and 'the' appear alongside each other we consequently remove the ⟨DET⟩ tag to produce the tagged sentence in (2.3) which contains the two chunks 'I love going' and 'to the cinema':

(2.3) ⟨PERS_PRO⟩ I love going ⟨PREP⟩ to the cinema

2.2.2 EBMT - an example

In order to illustrate how EBMT works let us first consider the task of training an EBMT system.

SENTENTIALLY ALIGNED CORPUS

Ich wohne in Dublin	⟷	I live in Dublin
Es gibt viel zu tun in Paris	⟷	There's lots to do in Paris
Ich gehe gern ins Kino mit meiner Frau	⟷	I love going to the cinema with my wife

Figure 2.1: Sententially aligned corpus, where we have the source language sentences on the left-hand side, and their aligned target-language equivalents on the right-hand side.

Given the sententially-aligned bilingual corpus in Figure 2.1, we see our source language German sentences on the left-hand side, and the target language English sentences on the right-hand side. From this corpus we are able to go beneath the sentence level and identify subsentential alignments which are then stored for later use in our phrasal database. We achieve this by using a set of closed-class words, or *marker* words, to segment the aligned source and target sentences into smaller chunks, and by using a number of similarity metrics to measure the most likely chunk sequence between source and target chunk sequences. For example, consider the two chunked sentences in (2.4):

(2.4) ⟨PERS_PRO⟩ I love going ⟨PREP⟩ to the cinema
↪ ⟨PERS_PRO⟩ Ich gehe gern ⟨PREP⟩ ins Kino

Based on a number of similarity metrics such as word co-occurrences, part-of-speech labels, relative chunk length, and cognate information (described in detail in section 3.4) we are able to compute the most likely alignments, as shown in (2.5):

(2.5) I love going $\longleftrightarrow$ Ich gehe gern
to the cinema $\longleftrightarrow$ ins Kino

Also from our original bitext we are able to extract word alignments which result in the formation of a bilingual dictionary, to which we resort when no sentence or phrasal examples are found. These word alignments are achieved through the use of the GIZA++ statistical word alignment toolkit (Och and Ney, 2003) detailed in section 3.2. Training of the system is now complete, as we now have an example database of aligned sentences, chunks, and words, as shown in Figure 2.2.

Now that our training is complete, let us consider the task of translating the following German sentence into English in (2.6):

(2.6) Ich wohne in Paris mit meiner Frau

Given all our aligned data in Figure 2.2, we start the translation process by searching the source side of our sententially aligned database to see if it contains the entire input string 'Ich wohne in Paris mit meiner Frau'. We do not find any such matches, so we move onto the next step of segmenting our input sentence into the smaller subsentential chunks: 'Ich wohne', 'in Paris', and 'mit meiner Frau'. We then search for these chunks on the source side of our subsentential example database, and see that we are able to find matches for each chunk. We then choose the corresponding target language examples as translation candidates, which are shown in Figure 2.3, and recombine them to produce a final translation as in (2.7).

(2.7) Ich wohne in Paris mit meiner Frau $\longrightarrow$ I live in Paris with my wife

ALIGNED DATA AFTER TRAINING

Sentence Alignments		
Ich wohne in Dublin	⟷	I live in Dublin
Es gibt viel zu tun in Paris	⟷	There's lots to do in Paris
Ich gehe gern ins Kino mit meiner Frau	⟷	I love going to the cinema with my wife

Phrase Alignments		
Ich wohne	⟷	I live
in Dublin	⟷	in Dublin
Es gibt viel	⟷	There's lots
zu tun	⟷	to do
in Paris	⟷	in Paris
Ich gehe gern	⟷	I love going
ins Kino	⟷	to the cinema
mit meiner Frau	⟷	with my wife

Word Alignments		
Ich	⟷	I
wohne	⟷	live
in	⟷	in
Dublin	⟷	Dublin
viel	⟷	lots
zu tun	⟷	to do
Paris	⟷	Paris
ins	⟷	to the
mit	⟷	with
meiner	⟷	my
Frau	⟷	wife

Figure 2.2: All our aligned data, which we were able to extract from our original aligned corpus in Figure 2.1. Source language examples appear on the left-hand side, and target language examples on the right-hand side.

2.3 Why EBMT over other methods?

There are a number of methods we could have chosen to produce foreign language subtitles. We weighed up the pros and cons of each, and finally settled on EBMT as our preferred approach for a number of reasons, which I outline in the following subsections.

2.3.1 EBMT versus translation memory

A translation memory (TM) is primarily designed to aid human translators by providing them with a database of text segments in a source language along with their

USEFUL SUBSENTENTIAL EXAMPLES

Ich wohne	$\longrightarrow$	I live
in Paris	$\longrightarrow$	in Paris
mit meiner Frau	$\longrightarrow$	with my wife

Figure 2.3: Useful examples were identified from our phrase alignments, and their target language counterparts extracted.

target language equivalents. The translator first supplies some source language text to be translated, which is then matched against the source side of the database and a target language output sentence presented to the translator for review. TM systems have generally been accepted by the translation community and are ubiquitous in large-scale translation companies, and are also used by the majority of freelance translators (Lagoudaki, 2006). Even though EBMT draws some parallels with TM, there is one essential difference: TM software needs a human present at all times during the translation process, and does not translate automatically. In other words, such systems can be said to be sophisticated search and replace engines (Schäler et al., 2003). EBMT, on the other hand, is an essentially automatic technique; having located a set of relevant examples, the system recombines them to derive a final translation, rather than handing them over to the human to decide what to do with them. We need our system to be fully automatic if we are to produce subtitles where they do not already exist. If need be, some post-processing can be done on the output to improve its quality.

Another major benefit of EBMT is that search goes below the level of the sentence to obtain subsentential examples, meaning we do not miss out on matches which may not be seen by looking at the sentence as a whole. However, the two paradigms are becoming more and more similar (Simard and Langlais, 2001), with second generation TM systems adopting a subsentential approach to extracting matches and postulating a translation proposal based on these matches.

2.3.2 EBMT versus rule-based machine translation

Rule-based systems generally rely on an extensive lexicon (complete with morphological, syntactic and semantic information) to parse the input text, creating an intermediary representation of the text. A set of complex rules, coded by hand, is then made use of to translate this symbolic representation into the target language. Some research has previously been carried out using rule-based machine translation (RBMT) for the translation of both closed captions (Popowich et al., 2000) and subtitles (Piperidis et al., 2005). However, it is difficult to measure the success of the rule-based systems used in these studies as no papers exist which deal with the evaluation of the RBMT-produced subtitles.

There is a widespread belief that rule-based systems will never be good enough to warrant serious consideration due to their lack of robustness and coverage, where lexicons are expensive to build, and where many rules will always have an exception. A pure rule-based system relies on an exact-match reasoning and fails to translate when it has no knowledge that matches the input text exactly. These systems must include a fail-safe mechanism which is capable of searching rules which can translate an expression similar to the input. Formulating rules for RBMT can be an extremely time-consuming, expensive and difficult process, and involves the work of linguistically trained experts. Improvement of translation quality of rule-based systems is based on the modification of these rules, where it is important to keep the consistency of mutually dependent rules. Although the use of generalized templates in EBMT is similar to the use of rules in RBMT (Somers, 2003), the traditional notion of EBMT does not involve any rules, and improvements to translation quality are made simply by adding appropriate examples to the database, and removing any *bad* examples.

2.3.3 EBMT versus statistical machine translation

Almost all research undertaken today in MT is corpus-based rather than rule-based. The two main corpus-based techniques are example-based machine translation, and statistical machine translation (SMT), with SMT systems being by far the more dominant. SMT relies on huge quantities of both monolingual and bilingual data using a range of theoretical approaches to probability distribution and estimation to deduce language and translation models. From the translation model we are able to establish a set of target language words (and more recently phrases) which are likely to be useful in translating the source language string. This is done by measuring source and target word/phrase co-occurrence frequencies, sentence length and relative sentence positions of source and target words. The language model then tries to assemble these words and phrases in the best order possible, by determining all bigram and/or trigram frequency distributions occurring in the training data.

However, the two paradigms have converged over the years, and it has never been more difficult to tell the difference between the two approaches. Up until recently SMT translation models were based on the the simple IBM word alignment models of Brown et al. (1990), but with today's phrase-based SMT systems (Koehn, 2003; Och, 2003) both SMT and EBMT approaches are capable of identifying useful subsentential chunks larger than the word. Despite the fact that EBMT systems have been modeling lexical and phrasal correspondences for over 20 years, no papers on SMT acknowledge this debt to EBMT.

Some advantages that EBMT has over SMT are as follows: EBMT alignments remain available for reuse in an example database, whereas SMT alignments 'disappear' in the probability models. In addition, SMT systems never learn from previously encountered data, so that when an SMT system comes across a string it has seen before it processes it exactly the same way as an unseen string. In contrast, EBMT will search for such strings in its example database, and output the

translation in a straightforward manner. Finally as SMT does not use any linguistic information to segment its sentences into smaller units, some of these chunks may contain words that are not necessarily connected, which may result in a number of unrelated chunk pairs.

2.4 Summary

In this chapter we began by documenting why we feel MT is well-suited to subtitle translation. We then went on to describe in detail the approach that we took to machine translation, which is example-based. EBMT is by no means the only translation technology solution we could have chosen and we accordingly compared and contrasted other methods to EBMT, namely: translation memories, RBMT and SMT.

Chapter 3

System Architecture

For our experiments in chapter 5 we use the recently developed corpus-based MT system: MaTrEx. The system was designed and implemented by a team of researchers at the National Centre for Language Technology (NCLT) in DCU. Although it was the first time that such a large-scale project was undertaken at the NCLT, it has proved to be huge success, with each person involved playing an essential part in the development process.

Programmed entirely in Java, MaTrEx is modular in fashion, which allows the user to extend and reimplement modules at ease. It is easily adaptable to new language pairs, and also allows one to adopt either an example-based or statistical approach to MT by changing the different chunking, alignment, and decoding modules. We take an example-based approach, where we based the core modules on the EBMT system developed by Gough (2005). MaTrEx is extremely efficient in terms of memory and is capable of handling large amounts of data in a short space of time. Having used the OpenLab initiative (Armstrong et al., 2006d) to debut the system, it has been used to conduct a number of experiments (Groves and Way, 2006; Stroppa et al., 2006), including our own (Armstrong et al., 2006a,b,c), and continues to be the main MT system used at the NCLT.

We begin this chapter by giving an overview of the system and its components in section 3.1. We then proceed by analysing each module in turn in sections 3.2–3.5, discussing in detail the alignment techniques used during the 'chunk alignment' stage, as well as the decoder utilised.

3.1 Overview of the System

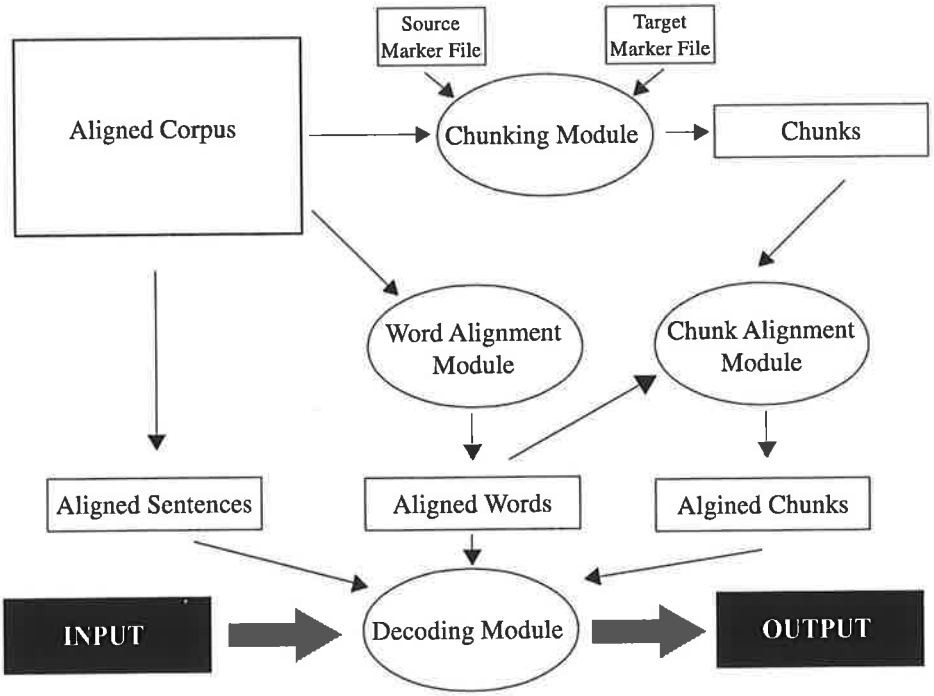

Figure 3.1: The MaTrEx System Architecture

In order for any EBMT system to produce a translation it must first be trained on an aligned bilingual corpus. Adapting the MaTrEx to work like an EBMT system, we start the training process by extracting a bilingual dictionary thanks to the **word alignment module**. The next step is to segment each aligned sentence in our training corpus into smaller 'chunks' using the **chunking module** and then computing the alignments between these chunks by way of the **chunk alignment module**. Training of the system is completed by combining all aligned sentences from our original bitext with the extracted chunk and word alignments to form an

example database. The **decoding module** takes in text as input, segments the input into smaller units, extracts suitable translation candidates from our example database and recombines these segments based on a language model. Figure 3.1 illustrates how the various system modules and data resources interact.

3.2 Word Alignment Module

The word alignment module takes in an aligned corpus as input and produces a set of word alignments and their probabilities. We make use of the GIZA++ statistical word alignment toolkit[1] (Och and Ney, 2003) to perform word alignment in both language directions, source-target and target-source, and obtain the union between these uni-directional alignments to increase coverage. This allows us to identify cases where a word in one language corresponds to more than one word in the other language. Compound nouns tend to be much more common in German than they are in English: *only child* ↔ *Einzelkind*, *main station* ↔ *Hauptbahnhof* and *special offer* ↔ *Sondernangebot* to name but three examples from our data.

```
# Sentence pair (13) source length 11 target length 13 alignment score : 4.55776e-18
mein guter , es tut mir leid , ich habe meinen schlüssel verloren
NULL ({ }) darling ({ 1 2 }) , ({ 3 }) i ({ 4 }) am ({ }) sorry ({ 5 6 7 }) , ({ 8 })
but ({ }) i ({ 9 }) lost ({ 10
13 }) my ({ 11 }) key ({ 12 })
```

Figure 3.2: Word alignments for the language direction English–German

Figure 3.2 shows a sample taken from one of our word alignment files. On the first line we have some statistical information, such as the length of the source and target sentences in words. Next comes the target sentence, and finally the source sentence is listed word by word, with references to source sentence in {}. For example, the first word *darling ({1 2 })* is aligned to the first two words in the target *mein guter*, and the fourth word *am ({ })* is unaligned.

[1] `http://www.fjoch.com/GIZA++.html` [Accessed October 2006]

When alignment is run in the opposite direction (Figure 3.3) it would be impossible to obtain the alignment: *mein guter* → *darling* as each target word may only be aligned to at most one source word. As we run alignment in both language directions, and obtain the union between alignment sets, we do not miss out on alignments such as: *darling* → *mein guter*.

```
# Sentence pair (13) source length 13 target length 11 alignment score : 1.42664e-15
darling , i am sorry , but i lost my key
NULL ({ 6 }) mein ({ }) guter ({ 1 }) , ({ 2 }) es ({ }) tut ({ 4 }) mir ({ 3 })
leid ({ 5 7 }) , ({ }) ich ({ 8 }) habe ({ } )
meinen ({ 10 }) schlüssel ({ 11 })
verloren ({ 9 })
```

Figure 3.3: Word alignments for the language direction German–English

3.3 Chunking Module

The chunking module takes in an aligned corpus and chunks this data based on a set of marker tags. Based on the *marker hypothesis* first mentioned in section 2.2.1, we tag each marker word in the training set with its corresponding marker tag. We have carried out several experiments (Way and Gough, 2005; Stroppa et al., 2006; Groves and Way, 2006) using this idea as the basis for the chunking component of our EBMT system, and found it to be a very efficient way of segmenting source and target sentences into smaller chunks. Sets of closed-class (or marker) words, such as determiners, conjunctions, prepositions, and pronouns, are used to indicate where one chunk ends and the next one begins, with the constraint that each chunk must contain at least one content (non-marker) word.

A sample of some of the English marker words and tags are shown in Table 3.1. For English and German we extracted marker words and tags from the lexical database CELEX (Baayen et al., 1995), and edited these manually in order to simplify the tags, and ensure that each marker word has only one marker tag. In total

we use 452 marker words for English and 560 for German. Some marker words may receive different parts of speech, for example, *her* may be appear as a possessive determiner as in *I read* ***her*** *book* or it can also appear as an object pronoun as in *I gave* ***her*** *a kiss.* We allow only one tag per marker word, and in cases such as this, the more common tag is chosen as this marker word's marker tag. Similarly, articles in German such as *der* can be nominative masculine as in ***Der*** *Mann ist schlecht*, or in the dative feminine *Ich habe das Buch* ***der*** *Frau gegeben.* The tags we use are general part-of-speech tags, and therefore these two sentences would both be marked by the ⟨DET⟩ tag without any reference to case or gender.

Determiners	⟨DET⟩	the, a
Quantifiers	⟨Q⟩	many, most
Prepositions	⟨P⟩	on, at
Conjunctions	⟨C⟩	and, but
Possessive Pronouns	⟨POSS_PRON⟩	mine your
Personal Pronouns	⟨PERS_PRO⟩	he, she
Verb to be	⟨BE_V⟩	am, are
Punctuation	⟨PUNC⟩	!, "

Table 3.1: Some of the English marker tags and words used during the chunking phase. Note that all tags denote the start of a new chunk except for the ⟨PUNC⟩ tag.

For a more detailed look at the chunking process, let us consider the English–German example in (3.1) taken from our training corpus:

(3.1) Darling, I'm sorry but I've lost my key

⟶Mein Guter, es tut mir leid aber ich habe meinen Schlüssel verloren

For the first step we automatically tag each closed-class word with its marker tag, as in (3.2):

(3.2) Darling ,⟨PUNC⟩ ⟨PERS_PRO⟩ I ⟨BE_V⟩ am sorry ⟨CONJ⟩ but ⟨PERS_PRO⟩ I've lost ⟨POSS_PRO⟩ my key

⟶⟨POSS_PRO⟩ Mein Guter, ⟨PUNC⟩ ⟨PERS_PRO⟩ es tut ⟨PERS_PRO⟩ mir leid ⟨CONJ⟩ aber ⟨PERS_PRO⟩ ich habe ⟨POSS_PRO⟩ meinen Schlüssel verloren

As every chunk must contain at least one non-marker word, we just keep the first marker tag when multiple marker-words appear alongside each other and discard the other tags (3.3):

(3.3) Darling ,⟨PUNC⟩ I am sorry ⟨CONJ⟩ but I've lost ⟨POSS_PRO⟩ my key
⟶Mein Guter,⟨PUNC⟩ es tut ⟨PERS_PRO⟩ mir leid ⟨CONJ⟩ aber ich habe ⟨POSS_PRO⟩ meinen Schlüssel verloren

3.4 Chunk Alignment Module

The chunk alignment module takes in source and target chunks produced during the chunking process (section 3.3), and produces a set of subsentential alignments. In order to determine alignments between chunks we use a dynamic 'edit-distance-like' algorithm (section 3.4.1). This algorithm is extended to allow for block movements, or jumps, and is incorporated to deal with differences that may arise between the order of constituents in source and target languages. Distances are calculated between each chunk in a sequence based on a combination of similarity metrics (section 3.4.2), and the most likely path is chosen between chunks.

3.4.1 Edit Distance Algorithm with Jumps

In the following, a denotes an alignment between a target sequence e and source sequence f, with $I = |e|$ and $J = |f|$. $\mathbb{P}(e|f)$ denotes the probability of e given f. Given two sequences of chunks, we need to calculate the most likely alignment $\hat{a}$:

$$\hat{a} = \underset{a}{\operatorname{argmax}}\, \mathbb{P}(a|e, f) = \underset{a}{\operatorname{argmax}}\, \mathbb{P}(a, e|f) \tag{3.4}$$

We first consider alignments such as those obtained by the edit-distance algorithm (Levenshtein, 1965; Wagner and Fischer, 1974). These consist of a set of

tuples, for example in (3.5):

$$a = (t_1, s_1)(t_2, s_2) \cdots (t_n, s_n), \tag{3.5}$$

with $\forall k \in [1, n]$, $t_k \in [0, I]$, and $s_k \in [0, J]$, and $\forall k < k'$:

$$t_k \leq t_{k^J} \text{ or } t_{k^J} = 0,$$
$$s_k \leq s_{k^I} \text{ or } s_{k^I} = 0,$$
$$I \in \cup_{k=1}^{n}\{t_k\}, J \in \cup_{k=1}^{n}\{s_k\},$$

where $t_k = 0$ denotes a non-aligned target chunk, and $s_k = 0$ a non-aligned source chunk.

We then assume the model in (3.6):

$$\mathbb{P}(a|e, f) = \Pi_k \mathbb{P}(t_k, s_k, e|f) = \Pi_k \mathbb{P}(e_{t_k}|f_{s_k}), \tag{3.6}$$

where $\mathbb{P}(e_0|f_j)$ denotes an "insertion" and $\mathbb{P}(e_i|f_0)$ denotes a "deletion" probability.

Assuming the parameters $\mathbb{P}(e_{t_k}|f_{s_k})$ are known, the most likely alignment is calculated by a simple dynamic-programming algorithm.[2] Moreover, this algorithm is approximated to allow for block movements or 'jumps', following the idea introduced by (Leusch et al., 2006). This adaption is necessary to allow for potential differences in the order of constituents in source and target languages (cf. Figure 3.4).

3.4.2 Calculating the Parameters

Instead of using an expectation-maximization algorithm to estimate these parameters, as commonly done when performing word alignment (Brown et al., 1993; Och

[2]This algorithm is actually a classical edit-distance algorithm (Jurafsky and Martin, 2000b) where distances are replaced by log-conditional probabilities.

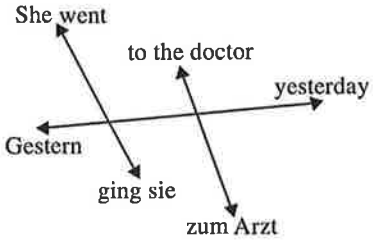

Figure 3.4: Equivalence between chunks in English and German

and Ney, 2003), we directly compute these parameters by relying on the information contained within chunks. In our experiments, we considered three main sources of knowledge: (i) word-to-word translation probabilities, (ii) distances based on chunk labels and (iii) distances based on the number of cognates per chunk. All three sources of knowledge are combined using a log linear model and are then stored as a single parameter to determine the relationship between two chunks.

Word-to-Word Probabilities

We make use of the Word-to-Word probabilities extracted during the word-alignment stage outlined in section 3.2.

Tag Information

Again we make use of the Levenshtein distance algorithm, this time to determine if two chunks have the same tag. If both source and target chunks contain the same marker tag, then a score of 1 is given, otherwise we give it a score of 0. Consider the chunk alignment candidate in 3.7:

(3.7) (a) ⟨PERS_PRO⟩ She said discipline ⟶⟨PERS_PRO⟩ Sie sagte Disziplin
(b) ⟨PERS_PRO⟩ She said discipline ⟶⟨BE_V⟩ ist überbewertet
(c) ⟨BE_V⟩ is overrated ⟶⟨BE_V⟩ ist überbewertet
(d) ⟨BE_V⟩ is overrated ⟶⟨PERS_PRO⟩ Sie sagte Disziplin

Candidates (a) and (c) will be given scores of 1, as both source chunks contain identical tags to their target match. Consequently candidates (b) and (d) will be given a score of 0. The total number of matching tags for a chunk pair is then stored as a parameter.

Cognate Information

Cognates have been employed for a number of bitext-related tasks, including sentence alignment (Simard et al., 1992), inducing translation lexicons (Mann and Yarowsky, 2001) and improving statistical MT translation models (Al-Onaizan et al., 1999). As English and German both originate from the same branch of the Indo-European language family, it is often the case that an English word and its German equivalent resemble each other on a lexical level: *active* $\leftrightarrow$ *aktive*, *discipline* $\leftrightarrow$ *Disziplin*, *university* $\leftrightarrow$ *Universität*, *wine* $\leftrightarrow$ *Wein*. Sometimes words are identical, such as: *arm* $\leftrightarrow$ *Arm*, *instrument* $\rightarrow$ *Instrument*, *radio* $\leftrightarrow$ *Radio*, *student* $\leftrightarrow$ *Student* and *zoo* $\leftrightarrow$ *Zoo*. In addition, many German words are used on a day-to-day basis in English: *Angst*, *Kindergarten*, *Rucksack*, *Wanderlust*, with same holding true for English words in German: *cool*, *party*, *sorry*, *ticket*. We use these examples for the basis for our assumption that if a chunk in the source language contains a word that closely resembles a word from its corresponding target language chunk candidate, there is a good chance that both chunks align to each other. However, a number of *false friends* exist between English and German (Table 3.2), where words may look the same but have different meanings. This means that relying on cognates alone for word alignments may produce mismatches.

In order to estimate the number of cognates a target chunk has with respect to the source chunk, we use a combination of the following methods:

- Longest common subsequence ratio (LCSR) (Hirschberg, 1977)
- Dice's coefficient (Dice, 1945)

FALSE FRIENDS

German false friend	English translation	Correct German term
bald	soon	bald = kahl
bekommen	to get	to become = werden
Dom	cathedral	dome = Kuppel
Fantasie	imagination	fantasy = Tagtraum
Gift	poison	gift = Geschenk
konsequent	consistently	consequently = folglich
Menü	today's special (restaurant)	menu = Speisekarte
Präservativ	condom	preservative = Konservierungsmittel
Puff	bordello	puff = Hauch
sympatisch	likeable / nice	sympathetic = mitfühlend
tasten	to touch	to taste = schmecken

Table 3.2: Some false friends between English and German.

- Minimum edit-distance ratio (MEDR) (Levenshtein, 1965)

The **LCSR** of two words is calculated by dividing the length of their longest common subsequence by the length of the longer word. For example, LCSR (*discipline, disziplin*) $= \frac{8}{10} = 0.8$, as their longest common subsequence is "d-i-s-i-p-l-i-n".

Dice's coefficient with respect to words, can be defined as the ratio of the number of shared bigrams to the total number of bigrams in both words. In the following, X denotes the set of bigrams for a source word, and Y the set of bigrams for a target word. Thus, Dice's coefficient, D, can be summed up as in 3.8:

$$D = 2 \times \frac{(|X \cap Y|)}{|X \cup Y|} \tag{3.8}$$

For example, *discipline* and *disziplin* share six bigrams (*di*, *is*, *ip*, *pl*, *li*, and *in*), so their Dice's coefficient is 2 x $\frac{6}{17} \simeq 0.71$.

The **minimum edit-distance** between two words is the minimum number of editing operations (insertion, deletion, substitution) needed to transform one word into the other, and is computed by a dynamic programming algorithm (Jurafsky and Martin, 2000b). We create a distance matrix (*distance*) with one column for each character in the target word, and one row for each character in the source

```
// function that takes in a source word and target word
// and returns the minimum edit-distance ratio
function minEditDistanceRatio(target,source) returns min_distance_ratio

    T = user-defined threshold
    n = length(target)
    m = length(source)

    Create a distance matrix distance[n+1,m+1]

    Fill the first column and the first row

    distance[0,0] = 0
    for i from 1 to n do
        distance[i,0] = i

    for j from 1 to m do
        distance[0,j] = j


    for each column i from 1 to n do
        for each row j from 1 to m do
            distance[i,j] = min(distance[i-1,j] + ins-cost(target_i),
                                distance[i-1,j-1] + subst-cost(source_j, target_i),
                                distance[i,j-1] + del-cost(source_j))

    min_distance = distance[n+1,m+1]

    if min_distance ≤ T
        then
            min_distance_ratio = min_distance / T
        else
            min_distance_ratio = 1
```

Figure 3.5: The minimum edit-distance algorithm (Jurafsky and Martin, 2000b)

word. Each cell in *distance[i,j]* contains the distance between the first i characters of the target and the first j characters of the source. Each cell can be computed as a simple function of the surrounding cells; thus starting from the beginning of the matrix it is possible to fill in every entry. The value in each cell is computed by taking the **minimum** of the three possible paths through the matrix, with the cell *distance(n+1,m+1]* giving the minimum edit-distance.

The algorithm is summarized in Figure 3.5, while Figure 3.6 shows the results

e	10	9	6	6	6	5	5	4	3	**2**
n	9	8	5	5	6	4	4	3	2	**1**
i	8	7	4	5	5	3	3	2	**1**	2
l	7	6	4	4	4	3	2	**1**	2	3
p	6	5	3	3	3	2	**1**	2	3	4
i	5	4	2	2	2	**1**	2	3	3	4
c	4	3	2	1	**1**	2	3	4	5	6
s	3	2	1	**0**	1	2	3	4	5	6
i	2	1	**0**	1	2	2	3	4	4	5
d	1	**0**	1	2	3	4	5	6	7	8
#	**0**	1	2	3	4	5	6	7	8	9
	#	d	i	s	z	i	p	l	i	n

Figure 3.6: The edit-distance matrix for the words *discipline* and *disziplin* using the minimum edit-distance algorithm of Figure 3.5, with a cost of 1 for insertions, deletions and substitutions.

of applying the algorithm the words *discipline* and *disziplin*. Looking at the top-right cell in Figure 3.6 we can see that the minimum edit-distance between the two words is *2*. Using an edit-distance of *3* as our threshold, we divide the minimum edit-distance by the threshold to give us our minimum edit-distance ratio (**MEDR**), which, for our example, is $\frac{2}{3} \simeq 0.66$.

The LCSR, Dice's coefficient, and MEDR are then averaged to give an overall probability of the target word being a cognate of the source. Again we need a threshold which determines whether the cognate probability is high enough to imply that two words are in fact analogous. This process is repeated for all possible word pairs in the source and target chunks, with the total number of cognates being added as a source of knowledge for our chunk information parameter.

3.4.3 Combining the Chunk Information

After we have gone through the process of determining relationships between a source and target chunk, it is necessary to combine these sources of knowledge to

get an overall likelihood score for the source f and target e chunk pair. It is possible to combine this similarity information in a log-linear framework in 3.9:

$$\log P(e_i|f_j) = \sum \lambda_k \log P_k(e_i|f_j) - \log Z, \tag{3.9}$$

where $P_k(.)$ represents a given source of knowledge, λ_k the associated weight parameter, and Z a normalization parameter, which is used as it is possible that $\sum \lambda_k \log P_k(e_i|f_j) = 0$. For example, suppose we want to determine the probability of the chunk-pair in (3.10):

(3.10) e = ⟨PERS_PRO⟩ I love going
f = ⟨PERS_PRO⟩ Ich gehe gern

Firstly, using word-to-word probabilities as our similarity metric we calculate $P_{wTw}(e|f)$ as 0.21. Next using marker tags as a means of comparing the two chunks, we observe that both share the same marker tag, and calculate $P_{tags}(e|f)$ as being 1. Now as we want to assign more weight to chunks with the same marker tags we multiply $P_{wTw}(e|f)$ by $\frac{1}{3}$ and $P_{tags}(e|f)$ by $\frac{2}{3}$. We then combine the two probabilities as follows: $\log P(e|f) = \frac{1}{3} \log P_{wTw}(e|f) + \frac{2}{3} \log P_{tags}(e|f) - \log Z = \frac{1}{3} \times (-1.56) + \frac{2}{3} \times (0) - 0.02 = -0.54$.

3.5 Decoding Module

The decoding module is capable of using the original bitext for aligned sentences, along with the words and chunks derived from the word and chunk alignment modules. As our example-based decoder (Groves, 2007) is not yet ready for implementation, the decoding module provides a wrapper around the well-established phrase-based decoder Pharaoh (Koehn, 2004).

Table 3.3 shows a example of what the Pharaoh manual refers to as a phrase-table, which acts as the main knowledge base of the decoder. The target string

PHRASE-TABLE

the ||| der ||| 1.0
a small house ||| ein kleines haus ||| 0.8
a tiny house ||| ein kleines haus ||| 0.2
that is ||| das ist ||| 0.7
this is ||| das ist ||| 0.3

Table 3.3: This is the format that Pharaoh demands from a phrase-table. Each line begins with the target string, t, is followed by the source string, s, and ends with the conditional probability, $P(t|s)$.

appears first, followed by the source, then the conditional probability associated to that source and target string pair. We combine our aligned words, phrases and sentences, along with their probabilities, to create this table, which Pharaoh consults to figure out how to translate source sentences into the target language. Consider the first line in the phrase table as in (3.11):

(3.11) the ||| der ||| 1.0

This entry means that the German word *der* will always be translated as the English *the* as $P(the|der) = 1.0$. Looking at the second and third lines of the table as in (3.12):

(3.12) a small house ||| ein kleines haus ||| 0.8
a tiny house ||| ein kleines haus ||| 0.2

we can see that it is more likely for *ein kleines haus* to be translated as *a small house* as it has a greater probability, 0.8 versus 0.2.

The probability cost that is assigned to a translation is based on four models: *phrase translation, language model, reordering model* and *word penalty.* The **phrase translation** model ensures that the target language and source language phrases are good translations of each other. The **language model** ensures that the the target language output is as fluent as possible. The **reordering model** allows for

reordering of the input sentence but with a higher cost each time a segment is re-ordered. Finally, the **word penalty** penalises candidates with too many or too few words. Each of these components can be given a weight which sets its importance. For our experiments we keep the default settings with each model being weighted equally, except for the *word penalty* for which we do not use. These parameters are contained in the Pharaoh configuration file, *pharaoh.ini*, shown in Figure 3.7.

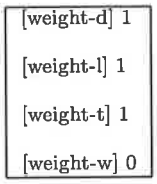

Figure 3.7: Pharaoh configuration file.

In order to run the decoder we use the command in (3.13):

(3.13)
```
% pharaoh -f pharaoh.ini -ttable-file phrase-table
-lmodel-file english.srilm < test.en-de.de > output.en
```

where *pharaoh.ini* is the configuration file, *phrase-table* where our words, phrases and sentences are stored, *english.srilm* the language model file, *test.de-en.de* the input file, and *output.en* the file that stores the translations. This computes translations for the language direction German–English.

3.5.1 Computing the Language Model

The SRI language modeling toolkit[3] (Stolcke, 2002) is used to create a language model file as required by Pharaoh. Statistical language models are used to estimate the prior probabilities of sequences of word strings, and have many applications in natural language processing: part-of-speech tagging, parsing, speech recognition, information retrieval as well as machine translation.

[3] `http://www.speech.sri.com/projects/srilm/` [Accessed October 2006]

For our system the language model is trained on the target language corpus, where n-grams are counted and probabilities derived. Based on these probability distributions, a language model is calculated, and is used in turn by the decoder to predict sequences of words and phrases in the output string. We use a trigram model, and Kneser-Ney discounting for n-grams to calculate the language model by calling the command in (3.14):

(3.14)
```
% ngram-count -text test-file -lm language-model-file -interpolate
-kndiscount1 -kndiscount2 -kndiscount3
```

3.6 Summary

In this chapter we have given a comprehensive description of the corpus-based MT system, MaTrEx, that we developed and adapted to deal with the task of translating DVD subtitles. We began by giving an overview of the system and its components. The system is based on a modular architecture, and we analysed each module in turn, particularly focusing on the techniques we used to align chunks such as cognate information, part-of-speech tags, and word-to-word probabilities.

Chapter 4

Corpora

In this chapter we document the various corpora we acquired for this study. As mentioned previously the MaTrEx system relies on a bilingual corpus for training purposes. In addition to the training corpus, we also needed to acquire suitable data to test the system. The training corpus needs to be aligned at sentence level for our system to extract suitable examples along with their translation correspondences.

Recent research has produced conflicting opinions in relation to which type of corpus gives the best results; Cavaglia (2002) discovered that for minimal sizes of training data, NLP performance improved when loaded with homogeneous data, whereas Denoual (2005) observed that a system loaded with heterogeneous data produced better results. MT output depends heavily on the nature of the task at hand, and as to the best of our knowledge no research has previously been carried out with respect to the use of homogeneous and heterogeneous data in the automatic translation of subtitles, it would be wrong to assume that one approach should be favoured over the other. Thus, we felt it important to gather together both homogeneous and heterogeneous data, and use this data to train the system in separate experiments and discover for ourselves which data works best for *our* test material. Acquiring this data was no mean feat and accordingly, we describe how the various corpora were obtained in section 4.1. A certain amount of pre-processing,

outlined in section 4.2, then had to be done to make the corpora MaTrEx-friendly. Finally we describe, compare and contrast the data in section 4.3.

4.1 Obtaining the data

Here we outline how our data for training and testing purposes was collected.

4.1.1 Ripping Subtitles: Homogeneous Data

First of all we explain exactly what we mean by homogeneous data. We assume that the source input for our system will be one or more subtitles from a wide variety of scenes, across many different genres of movie and television series. Although the type of language used in a movie is up to the discretion of the director, the subtitler has a less free role, and often has to conform to certain constraints that are the norm in the subtitle industry. Often a more simplistic syntactic structure is preferred over the original utterence. In addition to this, subtitles can frequently be seen as a transcription of spoken dialogue, and contain many examples of colloquialisms, contractions, and share many other traits of everyday speech. If the input to our system is to consist of subtitles from any genre, we make the assumption that a good example of homogeneous data would be another set of randomly generated subtitles and their translations. We found the best way for us to obtain this data was to build up a collection of DVDs which included both English and German subtitles, and then 'rip' these subtitles into text format[1]. DVD subtitles are stored as images on the DVD which are blended into the movie during playback.

It may be helpful to take a look inside a DVD to get an idea of its constituents. When you access the DVD drive you will see at least 2 directories: AUDIO_TS is used to store files for audio DVDs, and VIDEO_TS contains all the data needed for

[1]Since completion of this thesis it has come to our attention of a project relating to subtitle corpus collection (Tiedemann, 2007). For that piece of research, subtitles were obtained from the website `http://www.opensubtitles.org/en`, which contains subtitles ripped in srt format for over 3,000 movies and over 95 languages, which are freely available for download.

video including the menus, video, audio and subtitle streams. All this data is stored in 3 file types: .VOB, .IFO and .BUP (see Figure 4.1).

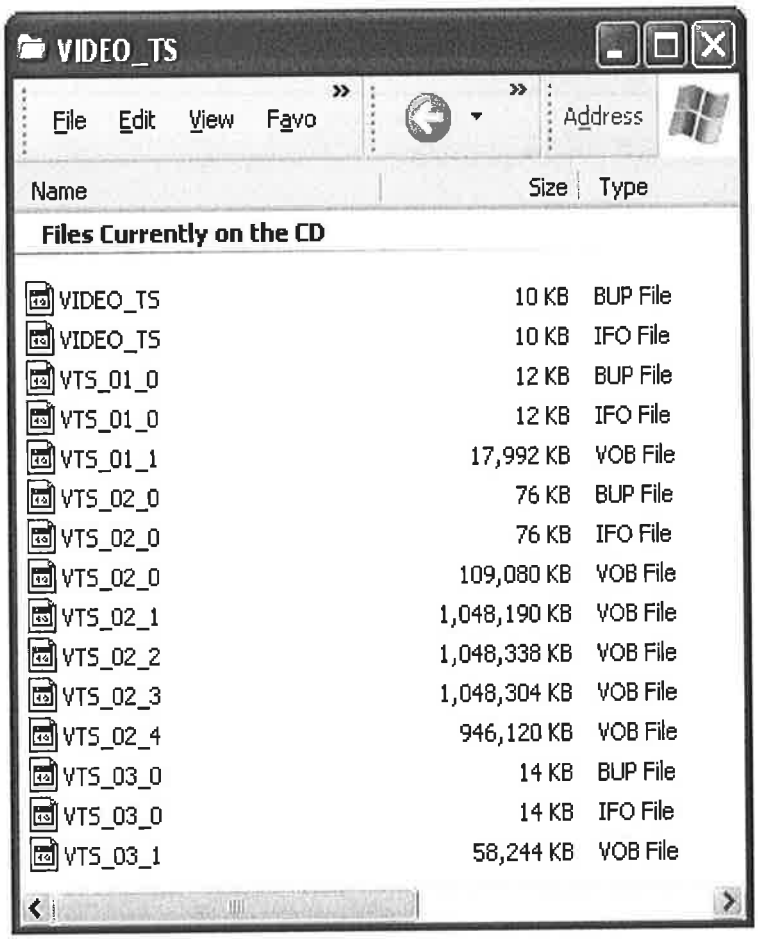

Figure 4.1: The VIDEO_TS folder of a DVD. VOB, IFO and BUP files are shown

A VOB file contains several streams multiplexed together: video, audio and subtitles. Video is MPEG-2, AC-3 is the standard for audio, and subtitles are stored as bitmap images. IFO files give the video player important navigational information such as chapter starts, and locations of the various audio and subtitle streams, and BUP files are just backups of these IFOs. We use **SubRip** (Zuggy, 2006) to convert the subtitles stored in the VOB to flat ASCII text. SubRip uses a similar technique to the optical character recognition (OCR) software used by scanners, where, with the help of the user, the images of the characters stored in the subtitle streams are

translated into a standard encoding scheme such as ASCII. Once the subtitles are in machine-readable format, with little pre-processing we can use them as direct input to our system.

First off in the ripping process, we start by opening up SubRip and navigating to the VIDEO_TS directory corresponding to the movie we want to extract our subtitles from. We then press *Open IFO* and select the IFO file that belongs to our ripped VOB files (by looking for the largest IFO file). Our next step is to choose which language stream we want to extract; we do so from the dropdown list in Figure 4.2. We make sure that *Subpictures to Text via OCR* is selected in the Action area, and also that the *Last Time Code* is set to 0:0:0.0 as SubRip only resets the time code after it has been closed and not after you have finished ripping subtitles for a movie. There is also a feature referred to as a *character matrix file*, which can be used to save the results of character recognition and use them to automatically decipher them later on; this can sometimes be useful when dealing with movies which were subtitled in the same font. Once that is all set up, we press *Start* to begin the character recognition process.

SubRip has to *learn* what each image represents, so after we start the recognition process we have to pay close attention to the characters we are entering, as if you make one mistake it will be seen throughout the final subtitle file. As time goes on and the character matrix starts to fill up you will notice that the processing speed of the subtitles increases significantly and less and less help is needed from the user. Occasionally SubRip will misinterpret certain phenomena as characters (such as borders and shading), but there is an option to change which text colours you want to process which may solve this problem. Another feature I found very helpful was the way in which you are able to assign shortcut buttons to your favourite characters, which is extremely useful when dealing with foreign character sets not easily accessible on the standard QWERTY keyboard.

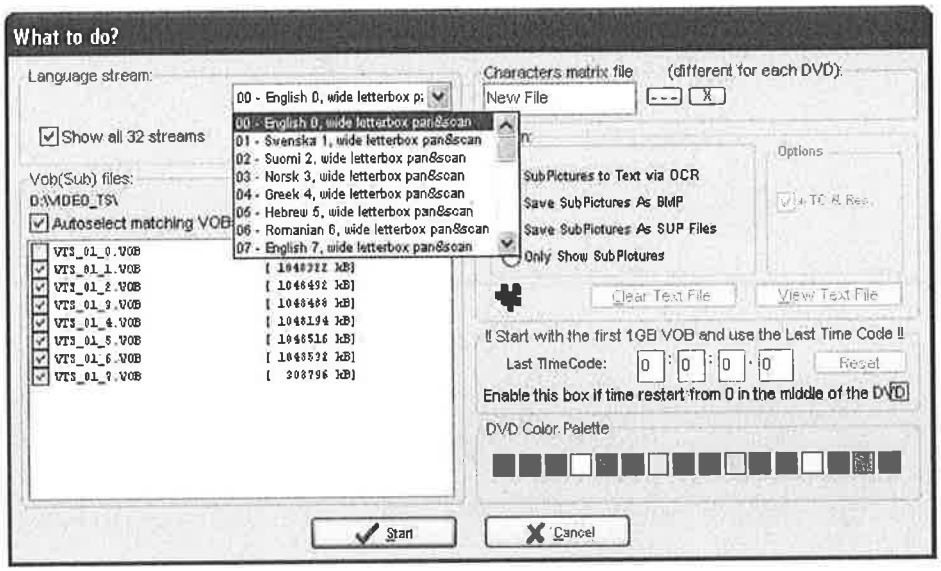

Figure 4.2: The SubRip start-up screen. Choose the subtitle language stream you want to process from the dropdown list. Here English is highlighted.

While SubRip is running you can get a preview of what the subtitle file will look like in the bottom window. Once processing is complete you simply save the subtitle file from the file menu. It is at this point that you may also find it practical to save the character matrix for later use. In total we ripped subtitles from over 50 movies, which were chosen from a wide range of genre such as: *action*, *comedy*, *thriller*, *film-noir*, *crime*, *science fiction* etc. Half of our movies we borrowed from the library and the other half were either borrowed from friends, already part of our own DVD collections, or purchased with funds from the project budget. These subtitles amounted to 42,000 sentence pairs.

4.1.2 Europarl: Heterogeneous Data

Our heterogeneous data comes from the Europarl corpus (Koehn, 2005) which is freely available for research purposes. The corpus itself was designed for the specific task of statistical machine translation in mind, with punctuation separated and

sentence boundaries clearly defined. It is available in many language pairs and has the added bonus of being already aligned at sentence level. We have used extracts from this corpus for a number of tasks here at the NCLT (Groves and Way, 2006; Stroppa et al., 2006; Armstrong et al., 2006a) and have shown that the MaTrEx system can cope well when trained and tested on such data (Armstrong et al., 2006d). Contrary to what some may think, there is not so much repetition in the Europarl, mainly due to the European Parliament dealing with different issues on a day-to-day basis. Thus, if we select sentences at random we should have our own heterogeneous corpus. For our experiments, we will test how much of an impact homogeneous and heterogeneous training data has on translation quality in separate experiments. In order for us to make a direct comparison between the two types of data, it was important to gather together equal amounts of training data. In total we took a random sample of 42,000 sentences of English along with their German translations from the Europarl corpus.

4.1.3 More Subtitles: The Test Data

In order to test the system, we randomly extracted over 2,000 sentence-pairs from our subtitle corpus, and used the source language sentences as input to the system. The automatic metrics we make use of in our experiments (see section 5.1) all make use of a set of reference translations to which the system output is compared to and the translation quality of the output estimated. We use the target language sentences from our test corpus as this set of reference translations. In addition to the data just mentioned, we also extracted subtitles from various DVD bonus material, which we used to test the capability of our system in handling material that may not normally be subtitled (these experiments are described in section 5.1.4).

4.2 Aligning the Subtitle Data

A few steps still had to be taken to convert our parallel corpus of subtitles into one that is also aligned sententially. We first narrowed down the search space for alignments, identified sentence boundaries, then calculated the most probable alignments based on sentence length and cognate information.

4.2.1 Narrowing down the search space

Like paragraph markers in normal text, chapter markers are used to split a DVD up into several sections. Using a program called **chapterXtractor** (Paris, 2002) we were able to determine chapter start and end times of our subtitle data, and with this information subcategorise the subtitles into several smaller files. This was done to help during the sentential alignment stage (see section 4.2.3), and also during evaluation when isolating subtitles for a particular scene (see chapter 5).

4.2.2 Sentence Identification

In order to mark what we defined as a sentence we used the *period*, *question mark*, *exclamation mark*, *colon*, *semicolon*, and *closing parentheses* as boundary indicators. Simple sentence disambiguation rules were applied; for example salutations such as *Mr.* and common abbreviated forms such as *etc.* were used as exceptions to the rule that a period means a new sentence; Table 4.1 includes some more examples. Periods can also be used as decimal points for which we used the regular expression `[0-9]*\ .[0-9]+` to identify such cases, and in internet addresses, which we identified with a number of regular expressions, for example, `(http://)?www\.(A-Za-z0-9)+\.com`. Another problem is that abbreviated forms are language-dependent, and we needed to have separate rules for both English and German.

SAMPLE OF ABBREVIATED FORMS

etc.	And so on; and so forth. From the Latin *et cetera*.
e.g	For example; for instance. From the Latin *exempli gratia*.
a.m.	Before noon. From the Latin *ante meridiem*
p.m.	After noon. From the Latin *post meridiem*
B.Sc.	Bachelor of Science.
M.P.	Member of Parliament.
Mr.	Mister.
U.N.	United Nations

Table 4.1: Table showing some of the abbreviated forms used to disambiguate between when a period denotes a new sentence or when it denotes an abbreviated form.

4.2.3 Sentential Alignment

Although we now had a bilingual corpus of subtitles, it was not yet aligned at sentence level. With the help of some of the extralinguistic features acquired during the ripping process (namely subtitle position, and time-code), and by using similar techniques to those used by Gale and Church (1991) in addition to our own outlined in section 3.4 on chunk alignment, we were able to produce a set of highly confident alignments. Subtitles and their translations act very much like sentence-pairs in any parallel corpus: they do not always align one-to-one. Often the subtitle translator ignores a source language subtitle, deeming it to be unnecessary (1-0 alignments), or feel that some extra information is missing from the source, such as something written in the background (0-1 alignment). Sentences may be longer in one language than in the other, and may cover several lines, and more than one subtitle, so it was also necessary to consider one-to-many and many-to-one alignments. Certain types of alignment are more common than others, so we weight their probabilities accordingly.

Figure 4.3 shows some ripped subtitles from chapter 7 of the movie: *Kill Bill Volume 1*. We measure the lengths of each line with respect to the number of characters and words per line. Next we calculate cognate information. Finally we take note of both the subtitle position and time-codes for the line. All these

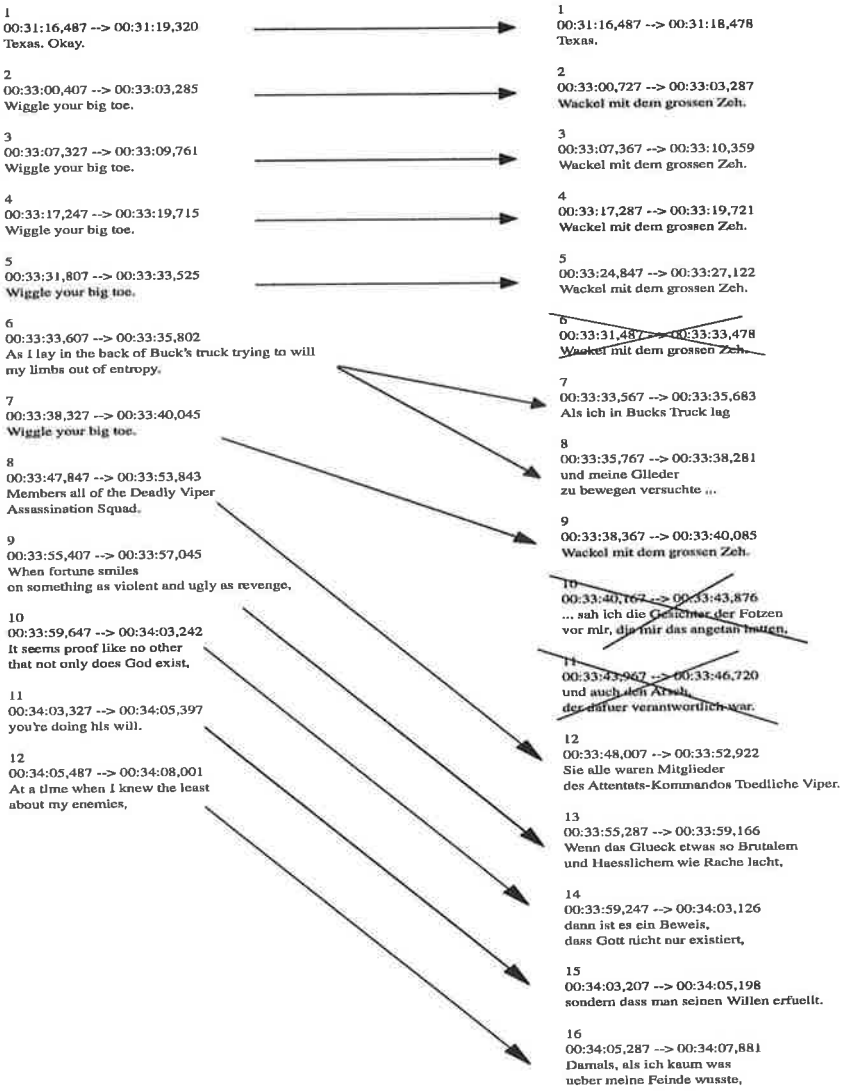

Figure 4.3: The alignments produced by our sentence alignment algorithm for a sample of source and target subtitles.

measures are then added to the feature vector for each particular line. Using the edit-distance algorithm we dynamically calculate distances for each feature-vector pair, and store the distances in a matrix. The shortest path is chosen and the most likely sequence of alignments produced. The alignments were then checked by hand for errors, which were few and far between, and the final result was that we now had a sententially aligned bilingual corpus of subtitles.

4.3 Describing, Comparing and Contrasting the Data

In order to get a better of idea of how our corpora compare to each other, in the following sections we analyse the data in relation to aspects such as lexis and syntax, and make comparisons based on these findings.

4.3.1 Basic Analysis

We begin by describing the data in terms of the number of sentences, tokens, and types, shown in Table 4.2. Like the sentence, there is no perfect algorithm for identifying word boundaries correctly one hundred percent of the time. However, it is important to be consistent with how you measure the number of words if you are comparing word frequencies across a number of corpora. We define a word-boundary as being any 'non-word-character', whether it be a space, a comma, period, or quotation mark, which is next to a 'word character' and vice versa. Therefore, anything that appears between a word-boundary we consider a token. The number of types in a text is equivalent to the number of distinct words. The *type-token ratio* (TTR) can be used as a somewhat crude measure indicating the vocabulary richness of a text (Luyckx et al., 2006) and is measured by calculating the ratio of types to tokens.

NUMBER OF SENTENCES, TYPES AND TOKENS

	sentences	tokens	types	TTR	STTR
Homogeneous Data	42000	226792	12399	5.47	39.62
Heterogeneous Data	42000	813297	19405	2.39	46.17
Test Data	2000	12197	2487	20.39	44.52

Table 4.2: Analysis showing the number of sentences, tokens and types, along with the type-token ratio (TTR) and standard type-token ratio (STTR) for the English training and test data

Looking at Table 4.2 we can clearly see that the TTR decreases significantly when dealing with increasing amounts of data. TTR varies dramatically with the

length of the text, and a 1,000 word article might have a TTR of 40%, a shorter one 70%, and a 4 million word article will most likely have a TTR of approximately 2%. TTR can be useful when dealing with relatively small amounts of text and comparing similar sized corpora, but when comparing different amounts of data a more advanced strategy is needed. In order to overcome this problem, we use what Scott (2006) calls the *standardised type-token ratio* (STTR). The TTR is calculated for the first 1,000 words, then calculated afresh at 1,000-word intervals to the end of the corpus. A running average is then computed, which means that you obtain an average TTR based on successive 1,000-word chunks of text. From the last column in Table 4.2 we can see that the STTR for the homogeneous data is 39.62% which, when compared to only 5.47% for the TTR for the same text, is a much more accurate measure of how lexically dense the text is.

SENTENCE LENGTH IN WORDS

	mode	median	mean	stdv
Homogeneous Data	10	10	12	7
Heterogeneous Data	36	38	39	17
Test Data	12	10	11	7

Table 4.3: Shows the mode, median, mean and standard deviation measures for the various English corpora with respect to the number of words per sentence

One of our hypotheses about the subtitle domain was that sentences are in general a lot shorter than sentences in other domains. We show this in Table 4.3 in terms of how sentence length differs across the various corpora with respect to the number of words each sentence contains. The mode is the most frequently occurring length, the median is the central length of the distribution, and the mean is the average of all lengths in the corpus. From the *mode*, *median* and *mean* measures we can conclude that sentences from the homogeneous data are of similar length to the ones we use for testing, and that our heterogeneous data sentences are on average 3 times as long when compared with the subtitle data. Furthermore, the standard deviation measures show that there seems to be a lot more variance in

sentence length for the heterogeneous data, with a standard deviation of 17 words as opposed to 7 words for both the homogeneous and test data.

4.3.2 Chi-square Test

Corpus similarity has been extensively studied in past literature, and a wide range of measures have been put forth. Kilgarriff (2001) compared different approaches such as Spearman, cross-entropy measures, and a chi-square test using word frequencies, with the latter proving to be the most suitable measure for comparing corpora. The chi-square test is a non-parametric statistical procedure which tests the relationship between the frequencies in a display. It allows the comparison of frequencies found experimentally with those expected on the basis of some theoretical model. If we employ the null hypothesis that there is no difference between the frequencies found in each category, the first step is to decide what the frequencies would have been had there been no relationship whatsoever between category and frequency. If this were so, all frequencies would be the same and equal to the sum of the frequencies in each cell, divided by the number of categories. This number of items is called the observed value, O. If O_1 and O_2 are the observed frequencies of a word in two separate corpora, C_1 and C_2, the expected value E is found using the formula in (4.1):

$$E = \frac{(O_1 + O_2) \times total\ tokens\ C_1}{total\ tokens\ C_1\ + total\ tokens\ C_2} \tag{4.1}$$

For example, consider calculating the expected value E for the word *YOU*, using the data in Table 4.4. We begin by adding O_1 and O_2, $468 + 9171 = 9639$. Next we multiply this by the total number of tokens of the first corpus, $9639 \times 12197 = 117566883$. Finally we calculate the total number of tokens in both corpora, $12197 + 226792 = 238989$, and divide to give us: $E = 117566883/238989 = 491.934$.

Next, for each observed frequency, the value $O - E$ is found and squared to give

more weight to the cases where the mismatch between O and E is greatest. Finally the value of chi-square is the sum of all the calculated values of $(O-E)^2$/E. Thus the formula is that in (4.2):

$$\chi^2 = \sum \frac{(O-E)^2}{E} \tag{4.2}$$

For all but random populations, $(O-E)^2$/E tends to increase with frequency. However, in natural language, words are not selected at random, and hence corpora are not randomly generated. If we increase the size of the data, we ultimately reach a point where all null hypotheses would be rejected. On this basis, Kilgarriff proposes a measure called chi by degrees of freedom (CBDF), and provides a method whereby a similarity based on data for more words should be directly comparable with one based on fewer words. Each chi-square value is divided by its degrees of freedom. The number of degrees of freedom is the number of cells in the frequency table minus one: for example in Table 4.4 a 6×2 contingency table was used, which means there are (6-1)*(2-1) = 5 degrees of freedom [2]. If the calculated chi-square value is greater than or equal to the critical value, we can dismiss the null hypothesis that the frequencies in the original table are similar to each other.

WORD FREQUENCY DATA

	Test Data	Homogeneous Data
Total Words	12197	226792
YOU	468	9171
I	435	7134
THE	370	7126
TO	288	5543
A	265	4919

	Test Data	Heterogeneous Data
Total Words	12197	813297
YOU	468	2507
I	435	10123
THE	370	27111
TO	288	14294
A	265	8464

Table 4.4: Contingency Tables showing word frequency

[2]The critical value for chi-square for a given significance level and degrees of freedom can be found in the appendix of (Thomas, 1996).

4.3.3 Lexical Analysis using CDBF

For this study we chose to compare the corpora both in terms of form (where we looked at the most common words which are predominantly closed-class, 'marker' words), and also in terms of content (where we look at predominantly open-class, 'lexical' words). Table 4.4 shows the frequencies for the 5 most common words in the test set *you*, *I*, *the*, *to* and *a* compared with the same words in the homogeneous and heterogeneous sets.

CALCULATION OF CHI-SQUARE FOR CLOSED-CLASS WORDS

	Test Data vs. Homogeneous Data					
	O_1	O_2	E_1	E_2	$(O_1\text{-}E_1)/E_1$	$(O_2\text{-}E_2)/E_2$
YOU	468	9171	491.9343	9147.066	1.164485	0.062627
I	435	7134	386.2901	7182.71	6.142147	0.330328
THE	370	7126	382.5645	7113.435	0.412655	0.022193
TO	288	5543	297.5899	5533.41	0.309035	0.01662
A	265	4919	264.5697	4919.43	0.0007	3.76E-05
Remainder	10371	192899	10374.05	192895.9	0.000898	4.83E-05
	Test Data vs. Heterogeneous Data					
	O_1	O_2	E_1	E_2	$(O_1\text{-}E_1)/E_1$	$(O_2\text{-}E_2)/E_2$
YOU	468	2507	43.9568	2931.043	4090.667	61.34766
I	435	10123	155.9986	10402	498.9901	7.483345
THE	370	27111	406.0426	27074.96	3.199347	0.047981
TO	288	14294	215.4548	14366.55	24.42649	0.366324
A	265	8464	128.9744	8600.026	143.4622	2.1515
Remainder	10371	750798	11246.57	749922.4	68.16544	1.022276

Table 4.5: Calculation of the chi-square for the top 5 most frequent words in the test data set and our training corpora

The chi-square statistic, with expected values based on probabilities in the joint corpus is shown in Table 4.5. Looking at the Test versus Homogeneous table, we sum the items in the last 2 columns to get a chi-square score of 8.46. Since a 6×2 contingency table was used, there were 5 degrees of freedom. We now consult the chi-square distribution table and see that the critical value (obtained by adding the values in the final two columns) on 5 degrees of freedom at the 99 percent significance level is 15.1. From this we accept the null hypothesis that the test data and subtitle data comprise of words drawn from the same population. Summing the items in the last two columns of the Test versus Heterogeneous table we get a chi-square score of over 5000. As we can see this score is skewed terribly by the fact that the word 'you' seldom occurs in the Heterogeneous corpus relevant to the Test

corpus, and when we apply the CBDF (used to correct the chi-square score for two considerably different sized corpora) we still get a high score of 980.3. This means that we can reject the null hypothesis that the Test and Heterogeneous Data come from the same population.

Next we are going to look at some open-class words. Scanning the top 500 most frequent words in the test data we chose the words *car*, *love*, *girl*, *dangerous* and *drink* which appeared well out of the range of other function words; *car*, the most frequent word chosen, appeared at position 142 in the top 500.

CALCULATION OF CHI-SQUARE FOR OPEN-CLASS WORDS

	Test Data vs. Homogeneous Data					
	O_1	O_2	E_1	E_2	$(O_1-E_1)/E_1$	$(O_2-E_2)/E_2$
CAR	13	134	7.502266	139.4977	4.028794	0.216671
LOVE	12	287	15.25971	283.7403	0.696325	0.037449
GIRL	11	163	8.880233	165.1198	0.506002	0.027213
DANGEROUS	5	27	1.633146	30.36685	6.941022	0.373292
DRINK	5	104	5.562905	103.4371	0.05696	0.003063
Remainder	12151	226077	12158.16	226069.8	0.04219	0.000227
	Test Data vs. Heterogeneous Data					
	O_1	O_2	E_1	E_2	$(O_1-E_1)/E_1$	$(O_2-E_2)/E_2$
CAR	13	35	0.709219	47.29078	212.9995	3.19435
LOVE	12	6	0.265957	17.73404	517.7066	7.764037
GIRL	11	3	0.206856	13.79314	563.1562	8.445643
DANGEROUS	5	66	1.049053	69.95095	14.88007	0.223156
DRINK	5	11	0.236406	15.76359	95.98654	1.439508
Remainder	12151	813297	12194.53	813132.5	0.155404	0.002331

Table 4.6: Calculation of the chi-square for open-class words

Using the CBDF method again, we get scores of 5.15 for the Test versus Homogeneous data, and 285.2 for Test versus Heterogeneous data, which means that we accept the null hypothesis for Test and Homogenous corpora, but reject it for the Heterogeneous corpora. Looking at individual words the chi-square score can give us a good indication of what words are 'key' to a text, i.e. what words are unusually high in one corpus when compared to another. For the words we looked at, the ones which were *key* in the test set compared to the heterogeneous set (the words where the chi-square score was higher than the critical value) were *you*, *I*, *to* and *a* from the closed-class words, and *car*, *love*, *girl* and *drink* from the open-class words. We have just looked at a few lexical examples here and to obtain a more detailed

description of where the corpora differ you would need to calculate the chi-square values for all words in the Test set and see how the corpus compares as a whole with the Homogeneous and Heterogeneous corpora.

4.4 Summary

In this chapter we have documented the corpora gathered together for our experiments. We have given a description of how to convert subtitles from image files into raw text, and shown how you can align this data to derive your own aligned corpus of subtitles. We went on to compare our corpora, and noted that sentences are generally shorter in our homogeneous data when compared with the heterogenous data from the Europarl corpus. Finally, using the chi-square test we were able to obtain a measure of how similar (or as in the case of the heterogeneous data, dissimilar) both training data sets are when compared to the test set. In addition, we explained how the chi-square test can be used to note what words are *key* when comparing one corpus to another.

Chapter 5

Experiments and Evaluation

Evaluation has played an integral part in the development of our system. Automatic metrics were used throughout as a quick and effective method of documenting what kind of impact certain changes to the system had on output quality. In addition to automatic evaluation metrics, we also conducted two types of human evaluation: (1) where participants were given randomly chosen system output in text format and asked to rate each output sentence with respect to intelligibility and accuracy, and (2) where we produced output for a selection of subtitles from different scenes within movies.

The first type of human evaluation was quite harsh as a random selection of subtitles were chosen which meant there was no context present. In addition no pre- or post-editing was done on the system output. However, it was still extremely useful as we were able to use the feedback from our participants to determine what our system does well and where it was going wrong. The second part of the human evaluation took place in a more viewer-friendly environment, which we felt was suitable to get an idea of whether the output produced by our system is at an acceptable level for public viewing.

In this section, we begin by describing how automatic metrics were used to

establish what type of training data produces optimal output: homogeneous data or heterogeneous data. Next we use the same automatic metrics to determine how well our system handles DVD bonus material. We then go on to give an account of how our two types of human evaluation were used to highlight what changes needed to be made in order to improve the system and also gauge the acceptability of our output.

5.1 Automatic evaluation

Although we can learn a lot from human evaluation, it can be an extremely time-consuming task and is really only appropriate for small-scale evaluations. On the other hand, automatic evaluation can be quick, language-independent and be applied to large-scale evaluations. Doddington (2002) observes that automatic scoring is at its most accurate when reference translations are of high-quality and the input sentences are from the same genre. As our test data consists of professionally-translated subtitles extracted from official DVD releases, we feel that our experiments can be considered to be well-suited to automatic evaluation metrics.

5.1.1 Metrics

For our experiments we used three well established automatic MT evaluation metrics: **BLEU** (Papineni et al., 2002), **NIST** (Doddington, 2002) and **WER** (Jurafsky and Martin, 2000a). All three metrics share in common a set of reference translations which are compared to the MT output to give a similarity score, but differ in the way in which these similarities are measured and scored:

- BLEU - Bounded between 0 and 1, where a **higher** score indicates a better translation. The geometric mean of the n-gram precisions is calculated with respect to a set of reference translations;
- NIST - Has a lower bound of 0, but theoretically no upper bound, where

higher scores indicate a better translation. NIST is a variant of BLEU but is based instead on the arithmetic mean of weighted n-gram precisions in the output with respect to a set of reference translations;

- WER - Has a lower bound of 0, but no upper bound, where a **lower** score indicates a better translation. WER or word-error rate is the edit distance in words between the system output and the reference translations.

5.1.2 Homogeneous versus heterogeneous training data - What is better for training?

The aim of these experiments is to determine what yields better results for the translation of subtitles: training an EBMT system on domain-specific data, or training on data from a different source. With almost all research in MT today being carried out using corpus-based techniques, it is strange to note that there has been somewhat few studies into the effect of the training-corpus on the final output of the system.

Until recently it was assumed that corpus-based MT systems achieve better results when trained on homogenous data. Denoual (2005) set out to reassess this general assumption for the language direction Japanese-English, and discovered that, contrary to this belief, his system yielded better results when trained on heterogeneous data, compared with equal amounts of homogeneous data. Using the BTEC corpus (a multilingual speech corpus comprised of tourism-related sentences), he randomly extracted 510 Japanese sentences and used these as input to the system. The system was then trained on increasing amounts of data (up to a maximum of 162,318 sentence-pairs) from the remainder of the corpus, and automatic evaluation metrics (BLEU, NIST and mWER[1]) were relied on to estimate the translation qual-

[1]Multi-reference word error rate (mWER) (Och, 2003) works in the same as WER, with the main difference being that the hypothesis translation is compared to the closest set of **multiple** reference translations.

ity of the output produced by the system. Based on these three measures, he shows that for increasing amounts of data, translation quality improves across the board. More notably, when trained on the random heterogeneous data, translation quality is found to be either equal or higher than when using homogeneous data for training: after performing a mean comparison of the 510 paired score values assigned to sentences, for instance at 50% of the training data, this difference is found to be statistically significant between BLEU, NIST, and mWER scores with confidence levels of 88.49%, 99.9%, and 73.24% respectively.

Denoual's findings prove true for larger amounts of data, but when trained on relatively small amounts[2], translation seemed to be of higher quality using the homogeneous data for training. No reason is given for this and it is unclear whether this cut-off point can be generalised for other types of corpora other than the sets he used during his experiments.

Subtitles can often appear very different from text in other domains; a quick glance at the statistics for our training corpora shows that sentences are much shorter in our homogeneous training data compared with sentences from the Europarl corpus (cf. Table 4.3). In addition, the χ^2 data (section 4.3.2) shows how statistically different our subtitle corpus is in comparison to the Europarl corpus. Each translation task is in itself unique, and as no previous research has been carried out with respect to the specific question of what training data is best for the task of automated subtitle translation, we believe that it warrants its own investigation.

Training and test sets

Overall we randomly extracted 40,000 sentence-pairs from our subtitle corpus and used this as our homogeneous data to train the system. The remaining 2,000 sentence-pairs from the subtitle corpus were then used as our test data. For our

[2]The exact figures are not mentioned but the threshold seems to lie at around 25,000 sentence-pairs.

heterogeneous training data we took 40,000 sentence-pairs at random from the Europarl corpus. Starting with 10,000 sentence-pairs, we trained the system on both the homogeneous and heterogeneous data in separate experiments, and calculated BLEU, NIST and WER scores on the same test set. These experiments were then repeated with increasing amounts of training data, in intervals of 10,000 sentence-pairs, until we had trained the system on a total of 40,000 sentence-pairs. The results for the language directions German–English and English–German are discussed below.

Results for German–English

AUTOMATIC EVALUATION RESULTS

		BLEU	NIST	WER
10,000	Homogeneous Data	0.1082	3.77	0.779
	Heterogeneous Data	0.0695	3.11	0.885
20,000	Homogeneous Data	0.1166	3.96	0.776
	Heterogeneous Data	**0.0740**	**3.21**	0.876
30,000	Homogeneous Data	0.1195	3.98	0.772
	Heterogeneous Data	0.0736	3.20	0.868
40,000	Homogeneous Data	**0.1287**	**4.08**	**0.761**
	Heterogeneous Data	0.0737	**3.21**	**0.865**

Table 5.1: Automatic evaluation results for the test set when loaded with increasing amounts of heterogeneous and homogeneous data: German to English

The results for German–English translation are shown in Table 5.1. From these results we can see that our system performs best when trained on 40,000 sentence-pairs of homogeneous data, that is our corpus of subtitles. When the full amount of 40,000 sentence-pairs of homogeneous data is used, we see a relative increase of 71.4% BLEU, when compared with equal amounts of heterogeneous data. In fact, we observe that training the system on 10,000 sentence-pairs of homogeneous data produces better results (46.8%relative BLEU) than when the system is trained on 4 times as much out-of-domain data. Also of note is that as we increase the amount of homogeneous data BLEU, NIST and WER results improve across the

board, with a relative increase of 10% at each 10,000 interval. This suggests that translation quality improves at a steady pace as we add more homogeneous data. The improvement in quality seems to be less significant with increasing amounts of heterogeneous data.

Results for English–German

AUTOMATIC EVALUATION RESULTS

		BLEU	NIST	WER
10,000	Homogeneous Data	0.0769	3.22	0.912
	Heterogeneous Data	0.0517	2.53	0.991
20,000	Homogeneous Data	0.0898	3.36	0.911
	Heterogeneous Data	**0.0581**	2.58	**0.984**
30,000	Homogeneous Data	0.1040	3.55	0.891
	Heterogeneous Data	0.0529	2.58	0.989
40,000	Homogeneous Data	**0.1088**	**3.58**	**0.856**
	Heterogeneous Data	0.0540	**2.59**	0.988

Table 5.2: Automatic evaluation results for the test set when loaded with increasing amounts of homogeneous and heterogeneous data: English to German

The results for a similar experiment but in the opposite language direction, English–German, are shown in Table 5.2. Although results are lower when compared with those in Table 5.1 (which is as expected due to there being many more cases of boundary friction when translating from English into German[3]), we actually see a greater difference between training the system on homogeneous and heterogeneous data: the maximum BLEU score is 0.108 which when compared with the maximum for the heterogeneous, 0.058, suggests a relative increase of 86% BLEU. Again we observe that BLEU, NIST and WER scores improve with increments of homogeneous data, and that training the system on smaller amounts of homogeneous data is still better than training on larger amounts of heterogeneous data: 10,000 of homogeneous data results in a 42% increase in BLEU when compared

[3]One example of boundary friction is in the translation from English into German of the determiner 'the', which can have many translations ('der', 'die', 'das', 'den', 'dem') depending on the words which surround it.

with 40,000 sentence-pairs from the Europarl corpus.

Summary of results

It is clear from these experiments that the type of corpus used to train the system has a serious impact on translation quality. We note that our system performs consistently better when trained on homogeneous data, when compared with equal or even greater amounts of heterogeneous data. Increasing the amount of homogeneous data also improves results across the board. However, we only had 40,000 sentence-pairs available for training purposes, so it would be interesting to obtain more homogeneous material, and see if translation quality continues to improve with larger amounts of this training data.

5.1.3 Choosing an optimal chunk alignment strategy

As mentioned previously in section 3.4 on chunk alignment, MaTrEx is capable of utilizing a number of chunk similarity metrics to determine the distance between chunks: cognate-based chunk distance, tag-based chunk distance, word-to-word probabilities, and distances based on the number of characters or words per chunk. In order to decide which of these metrics produces the best output for our data, we ran several experiments in which different combinations of the metrics were used, and BLEU, NIST and WER scores were calculated to obtain an estimate of the translation quality of the output. These results were then compared against each other, with the similarity metric combination with the highest average score being chosen as our 'optimal' chunk alignment strategy.

Experimental set-up

We used the same 40,000 sentence-pairs from our homogeneous data in section 5.1.2 to train the system, again using the remaining 2,000 sentence-pairs to test the system. We began by aligning chunks based on cognates, marker tags, word-to-word

probabilities, and the lengths of chunks in separate experiments. Output was produced using the chunk alignments produced by these strategies, and BLEU, NIST and WER scores were calculated. The same process was then repeated for combinations of the various chunk similarity metrics, again recording automatic evaluation results. These experiments were performed in both language directions: English–German, and German–English. Note that we did not incorporate word alignments at the final stage of decoding as we were only concerned with the performance of our chunk alignment strategies. This explains why the evaluation scores are lower when compared with the results obtained in section 5.1.2.

Results for German–English

AUTOMATIC EVALUATION RESULTS

Chunk Similarity Metrics	BLEU	NIST	WER
cognates	**0.0858**	**3.0131**	0.8989
marker-tags	0.0845	2.9860	0.8890
word-to-word probs	0.0849	3.0067	0.8879
length(chars)	0.0856	2.9863	0.8918
length(words)	0.0842	2.9868	**0.8820**

Table 5.3: Automatic evaluation results for German-to-English, using a number of chunk-alignment strategies

What we can note straightaway from the results in Table 5.3 is that the difference in automatic evaluation scores seems to be marginal across all chunk alignment strategies. According to BLEU and NIST scores cognate information performs best, but is only an increase of 0.02% absolute over the next best strategy: length based on characters. If we use WER as our evaluation metric of choice we see that the basic method of counting the number of words produces a slightly better score, an increase of 0.59% absolute over the next best strategy based on word-to-word probabilities. Our next step was to determine whether combining these different strategies resulted in a more significant improvement.

Looking at Table 5.4, we obtained best results when we used a combination of

AUTOMATIC EVALUATION RESULTS

Chunk Similarity Metrics	BLEU	NIST	WER
cognates + word-to-word probs	0.0851	3.0196	89.06
cognates + marker-tags	0.0872	3.0236	88.62
cognates + length(chars)	0.0885	3.0246	89.15
cognates + length(words)	0.0840	2.9978	89.55
word-to-word probs + length(chars)	0.0869	3.0221	88.60
word-to-word probs + length(words)	0.0843	3.0075	88.01
length(chars) + length(words)	0.0849	2.9755	89.31
cognates + word-to-word probs + length(chars)	**0.0892**	**3.0530**	**88.45**
cognates + word-to-word probs + length(words)	0.0836	2.9993	89.29
cognates + length(chars) + length(words)	0.0878	3.0260	88.87
word-to-word probs + length(chars) + length(words)	0.0869	3.0221	88.60

Table 5.4: Automatic evaluation results for German-to-English, using *combinations* of a number of chunk-alignment strategies.

cognates, word-to-word-probabilities and length based on characters, which shows a relative increase of 3% BLEU when compared with our best BLEU score for the single evaluation metrics in Table 5.3.

Results for English–German

Similar experiments were repeated for the language direction English–German, with the results being documented in Tables 5.5 and 5.6.

AUTOMATIC EVALUATION RESULTS

Chunk Similarity Metrics	BLEU	NIST	WER
cognates	0.0683	2.5619	1.0283
marker-tags	0.0685	2.5623	1.0196
word-to-word probs	0.0695	2.6108	1.0163
length(chars)	**0.0728**	**2.6493**	**1.0156**
length(words)	0.0711	2.6430	1.0162

Table 5.5: Automatic evaluation results for English-to-German, using a number of chunk-alignment strategies

This time round the difference between using the various similarity metrics was significantly greater, with a relative increase of 6% between best and worst BLEU scores. We observe that our strategies based on the number of characters and words per chunk, actually perform better than those based on more complex measures such

as cognate information and word-to-word probabilities. We then set out to see if we could improve on these results by combining these metrics as shown in Table 5.6.

AUTOMATIC EVALUATION RESULTS

Chunk Similarity Metrics	BLEU	NIST	WER
cognates + word-to-word probs	0.068	2.5624	1.0261
cognates + marker-tags	0.0695	2.5503	1.0255
cognates + length(chars)	**0.075**	**2.6671**	**1.0048**
cognates + length(words)	0.0705	2.6003	1.0185
word-to-word probs + length(chars)	0.0747	2.6616	1.0113
word-to-word probs + length(words)	0.0729	2.649	1.0061
length(chars) + length(words)	0.0738	2.6426	1.0166
cognates + word-to-word probs + length(chars)	0.0747	2.6616	1.0113
cognates + word-to-word probs + length(words)	0.0709	2.6149	1.0167
cognates + length(chars) + length(words)	0.0745	2.6498	1.0163
word-to-word probs + length(chars) + length(words)	0.0741	2.6537	1.0152

Table 5.6: Automatic evaluation results for English-to-German, using *combinations* of a number of chunk-alignment strategies

Analysing the data in Table 5.6 we observe that our system performs best when using a combination of cognates and length based on the number characters per chunk for the language direction English–German. Using this combination results in a 10% relative increase in BLEU when compared with using cognates alone. Furthermore, the best NIST and WER scores are achieved with these similarity metrics, which suggests that this is very likely to be our optimal combination. In contrast to the language direction German–English, this time round we find that combining more than two similarity metrics does not seem to improve translation quality. In fact, in certain cases using a single strategy produces better scores than using a combination of 3 metrics; compare the BLEU score for length based on characters (0.0728), and the combination: cognates + word-to-word probs + length based on characters (0.0709). As we have seen in section 3.2, source-target and target-source alignments may appear different, which could go some way to explaining why the same alignment combination does not produce the same results for both language directions.

5.1.4 Bonus material

One of the major selling points of DVDs is the ability to include bonus material, which may come in many different forms including director and cast commentaries, documentaries on the making of the movie, or interviews with cast members to name but a few. It is becoming increasingly popular to include subtitles for this material, but it is often the case that this bonus material is only subtitled in the original language of the DVD release with foreign language subtitles being ignored. One of the major aims of the project was to provide subtitles where they do not already exist. Thus, we set about seeing how well our system was able to deal with subtitles from a sample of bonus material in the following experiments.

Experimental set-up

From the fifty DVD releases we had at our disposal, only four contained both English and German subtitles for the bonus material. We used these subtitles to test the system in four separate experiments and calculated BLEU, NIST and WER results accordingly. Again we performed translation in both language directions. In order to train the system we used a random set of 40,000 sentence-pairs from our homogeneous corpus, and calculated chunk alignments thanks to our optimal chunk alignment strategies outlined in section 5.1.3. Our test data consisted of four sets of English subtitles and their German equivalents extracted from the bonus material for the following movies: *Harry Potter and the Philosopher's Stone*, *The Lord of the Rings: The Fellowship of the Ring*, *The Ring* and *American Graffiti*.

Results for bonus material

From the results in Table 5.7 we observe that, again as expected due to boundary friction, translation quality is best when translating from German into English, with the system performing on average 43% higher when compared with the same data in the opposite language direction. The difference in BLEU score is quite large for some of the bonus material: we see that BLEU score increases by over 100% when

AUTOMATIC EVALUATION RESULTS

		BLEU	NIST	WER
American Graffiti	German–English	0.1119	3.8063	78.14
	English–German	0.0716	3.0029	90.38
Harry Potter	German–English	0.0838	3.5323	76.52
	English–German	0.0701	3.0812	90.09
The Lord of the Rings	German–English	0.0748	2.8118	86.47
	English–German	0.0423	2.2518	104.93
The Ring	German–English	**0.1671**	**3.7603**	**70.92**
	English–German	0.1388	3.4918	78.95

Table 5.7: Automatic evaluation results for the bonus material. Translation was performed in both language directions German–English and English–German.

we compare the German–English results for *The Ring* and *The Lord of the Rings*. When we analysed the bonus material for *The Lord of the Rings* we found that a lot of the material came in the form of director and cast interviews. The bonus material for *The Ring* on the other hand contained a mini-feature where they borrowed heavily from the language used in the main feature. We used subtitles from this main feature to train the system, which would explain the relatively high BLEU, NIST and WER scores for *The Ring*.

What these results suggest is that our system handles bonus material to different degrees of effectiveness. We achieved the highest score yet by our system when it was tested on the bonus material for *The Ring*. The output for both *American Graffiti* and *Harry Potter* received similar results to when we loaded the system with random sets of homogeneous data. However, we also achieved some relatively low scores for *The Lord of the Rings* which indicates that the language used in certain bonus material can appear much more varied than with regular subtitles. In order to improve on these results it might be helpful to train the system on more subtitle data, including already existing subtitles for bonus material.

5.2 Human evaluation

Automatic evaluation is a quick, easy and often reliable way of estimating the translation quality of the output. However, none of the metrics mentioned previously make use of linguistic information, and are mainly concerned with either counting n-gram matches (BLEU and NIST), or calculating the edit distance between sentences based on words (WER). We felt it important to elicit a human's opinion on our output and therefore conducted two separate studies, where participants were given our output in different forms. Our human evaluation was essentially split into two types: *formative* and *summative*. Formative evaluation takes place during the development process, and is used to detect potential problems before the system is actually implemented (Preece, 1993). In contrast, summative evaluation is carried out when the system is finished (ibid). For the first evaluation study we used a 'no context' evaluation approach where participants were given the input used to test the system along with the translations the system produced, without actually seeing the movies from which these subtitles were taken. They were then asked to rate the system output with respect to intelligibility and accuracy. For the second evaluation study, we conducted a pilot study within an audiovisual environment in order to determine whether the output produced by our system was acceptable for public viewing.

5.2.1 Human evaluation - 'no context' approach

The aim of this type of evaluation was to determine where improvements could be made to the system. In order to estimate translation quality, we asked participants to rate a randomly generated set of output sentences according to the intelligibility and accuracy scales shown in Table 5.8. Intelligibility scores were determined first where the participant was just given the output sentences and asked to give each one a score from one to four. They were then given the sentences which were used as input to the system to produce these output sentences, and were asked to rate each

output sentence with respect to its accuracy to the corresponding input sentence. Intelligibility scores are useful as a translation may not always resemble the source text yet still be perfectly understandable. Accuracy scores are used to measure the closeness of the translation to the original source language sentence.

	INTELLIGIBILITY SCALE
1	Easily Comprehensible
2	Comprehensible
3	Difficult to comprehend
4	Incomprehensible
	ACCURACY SCALE
1	Output sentence fully conveys the meaning of the source sentence
2	On the whole, the output sentence conveys the meaning of the source sentence
3	Output sentence does not adequately convey the meaning of the source sentence
4	Output sentence does not convey the meaning of the source sentence

Table 5.8: The scales and the range of scores possible for the first human evaluation. These scales are based on work by van Slype (1980) and by Nagao (1984) as described in Jordan et al. (1993).

Experimental set-up

From our test set we extracted 200 sentences at random, and split these into four groups of 50 sentences. These were used as input to the system, which we trained on a random set of 40,000 sentences of homogeneous data. Five native speakers of English (with fluent German) were asked to evaluate the German–English output, with five native speakers of German (with fluent English) being given the English–German output to evaluate. The participants in the evaluation were first given the output produced for their mother tongue and asked to score each sentence for intelligibility. They were then given the source language set used as input to the system, and asked to compare this with the output to give an accuracy score for the translation. According to Kenny (2006) "it should be possible to evaluate intelligibility without any reference to the source text, so accuracy should not come into it; a text

can be completely intelligible but bear little resemblance to the source text, accuracy, on the other hand, should be ascertained independently from intelligibility".

Results

HUMAN EVALUATION RESULTS

		Intelligibility	Accuracy
	German – English		
40,000	Homogeneous Data	2.45	2.98
40,000	Heterogeneous Data	2.51	3.05
	English – German		
40,000	Homogeneous Data	2.2	2.65
40,000	Heterogeneous Data	2.7	2.8

Table 5.9: Average intelligibility and accuracy scores for random sentences of system output when trained on 40,000 sentence-pairs of homogeneous and heterogeneous data.

Table 5.9 shows average intelligibility and accuracy scores for the system trained on homogeneous and heterogeneous data. We note that average intelligibility accuracy scores suggest that, again, training the system on homogeneous data produces best results, and that in fact translation quality scored higher for the language direction English–German. This could be due to the subjective nature of the study, and because the number of participants are quite low (only 5 participants for each language direction), it is probably more useful to analyse the results in a qualitative and interpretive framework rather than quantitatively.

We chose to focus on different areas, where we identified high-scoring sentences as being good examples of what our system does well. Analysing output sentences with lower scores we were able to identify lexical errors as well as chunk mismatches.

Figure 5.1 presents an example where the system produced a suitable translation, even though it did not match the reference translation on the official DVD release. As the output bears no resemblance to the human-translated German equivalent, the automatic metrics used in our other experiments would have given this output

EBMT Output:	Scht, scht, scht! Gut Kinder, mehr Ruhe!
Input sentence:	Shh, shh, shh! Alright children, now quiet!
Reference translation:	Okay, Kinder, nun seid ruhig.

Figure 5.1: A sample sentence produced by our system, along with the sentence used as input, and the original translation of the source language sentence by a human translator.

a BLEU and NIST score of zero. However, we found the sentence to score quite highly with respect to the human evaluation. We see that the system has suitably translated the English utterance *shh* as *scht* and instead of the adjectival form of *quiet*, as chosen by the human subtitler, the system opted for the nominal form. Both German examples make sense when interpreted without any context, although the EBMT-produced subtitle may benefit more from the contextualisation offered by the extra semiotic channels present in an audiovisual environment. Judging from the data in Table 5.10, this sentence also provides evidence of the subjective nature of this type of evaluation. Out of the three people who were given this subtitle to evaluate, two gave it a perfect score for intelligibility, with the other person giving it a score of three.

	Participant 1	Participant 2	Participant 3
Intelligibility Score	1	1	3
Accuracy Score	3	3	3

Table 5.10: Human evaluation scores for the output subtitle: "Scht, Scht, Scht! Gut Kinder, mehr Ruhe!"

An example of where our system actually improves on the originally translated sentence is the translation of *I got the suitcase* into German. The original German subtitle is translated as *Ich habe den Koffer*. The EBMT translation produces the more colloquial *Ich hab den Koffer*. This may be seen as a more accurate translation of the input sentence as the English sentence uses the colloquial term *I got* rather than *I have*. Figure 5.2 shows some examples of some other phrases that our

system correctly translates in both language directions even though they are not literal translations of each other.

What's the matter?	⟷	Was ist los?
I don't know	⟷	Ich weiß nicht
Watch out!	⟷	Achtung!
Stand up	⟷	Steh auf
Twice as	⟷	Doppelt so
I'm going to try to	⟷	Ich werde es versuchen
I'm going home	⟷	Ich gehe nach Hause
We'll call later	⟷	Wir rufen später wieder

Figure 5.2: A sample of correctly translated phrases produced by our system.

Given that this evaluation was conducted at an early stage in development we were made aware of several errors that the system produced with relation to chunk-alignment, the recombination stage and also in translating word-for-word. Examples of these errors are shown in Figure 5.3 with the ideal translation on the left-hand-side and the machine-produced output on the right-hand side. As far as lexical errors go, this human evaluation helped us address cases where words were being mis-translated and also where no translation was available. Here we see how the system treats words that do not exist in the training set, such as the English word 'wide' and the German word *Kaviar*. We also note how *der größter* is used incorrectly, when the adjective *toll* would have been a more appropriate translation. Errors in the chunk-alignment process also became evident, where certain commonly used chunks were being mismatched. Here we see that the English *I hate* is incorrectly translated as *wie ich* instead of *I hate*, and that *I'm leaving my husband* is mistakenly translated as *Ich hinterlasse meinen Mann* which actually means *I'm leaving my husband* ***behind***. All these criticisms were taken on board and steps were taken to fix the errors found during the evaluation process.

doppelt so groß graben ⟷	doppelt so groß wide	no translation for 'wide'
champagne and a tin of caviar ⟷	champagne and Kaviar	no translation for 'Kaviar'
for a price ⟷	for money	incorrect chunk alignment
Du bist toll ⟷	Du bist der größter	incorrect chunk alignment
I hate ⟷	wie ich	incorrect chunk alignment
I'm leaving my husband ⟷	Ich hinterlasse meinen Mann	incorrect word alignment

Figure 5.3: Some mistakes made by the system that we were able to identify as a result of the evaluation and remove from our phrasal database.

5.2.2 Human evaluation - pilot study

For the next stage of the evaluation process we developed and conducted a pilot study into the acceptance of the subtitles produced by our system. This pilot study comes as a precursor to a full-scale study, which will be conducted at a later date by other project participants in their PhD. work when more improvements have been made to the system and the questions and methods used during our evaluation have been honed.

Experimental set-up

Our evaluation sessions took place in a newly renovated language research lab in the School of Applied Language and Intercultural Studies (SALIS), where the participants watched a total of six DVD clips on a 32inch flat-screen television. Six participants took part in the study, three male and three female, with each being fluent speakers of German, and all having a good knowledge of English. They were each shown the same 6 DVD clips which were subtitled by our system into German. After watching the clips each participant was asked a number of questions relevant to our research such as their attitudes to subtitles and translation technology, the accuracy of the EBMT subtitles, the aesthetics of and timings of the EBMT-produced subtitles, the influence source and target language knowledge had on their perception of the subtitles, and possibility of future applications of our system.

Attitudes towards Subtitling

Of the six participants, five were from Germany, with one from Austria. Both countries are traditionally dubbing countries. However, all participants said they watch subtitled movies at least once a month and that they actually prefer subtitles over dubbing when they have some knowledge of the original language of the movie. It was pointed out by one of the participants that "when films are dubbed they lose some of the original meaning and have less of an impact on the viewer as you don't get to hear the actor's real voice". A common question in the area of subtitling is whether or not the subtitler should translate everything that is said in the original script or produce a more condensed version, while still conveying the original meaning. Five of the participants agreed with the latter scenario saying it saves time when reading the subtitles.

Attitudes towards EBMT-generated subtitles

When asked to rate the overall accuracy of the subtitles produced by our system, they received an average score between *2* and *3*, based on the scales outlined above in section 5.2.1. A score of *2* means that 'on the whole the subtitles conveys the meaning of the original source language utterance', and a score of *3* means that 'the subtitles do not accurately convey the meaning of the original source language utterance'.

Although all participants said that they believed an automated approach should not be introduced when subtitling full-length features, several other scenarios were mentioned in which EBMT could be used. One participant, who works in software localisation, felt that with improvements to the lexicon the software could be useful in translating nature documentaries and crime series and in other types of programmes where the language is repetitive. Another participant felt that if subtitlers were to post-edit our system output it would speed up the subtitling process, which is what we also envisioned.

The aesthetics and timing of the EBMT-generated subtitles

Participants were asked to rate the appearance of the subtitles, where all bar one felt that the colour and size of the subtitling font was adequate. This one participant said she needed time to get used to subtitles which she felt were "a little squashed". All participants were happy with the placement of the subtitles and said that the splitting of each subtitle onto different lines was not an issue.

The subtitle timing was considered to be too fast by most participants, with some saying the reason for this was the time they needed to think about the subtitle due to a lexical error. The exposure times also presented difficulties for some, where they noticed that a small amount of text was on screen for what seemed like too long a time.

Influence of source and target languages

As mentioned all participants have knowledge of English, which ranged from good to excellent based on a self-assessment of language competence. All participants were able to identify mistakes in translation, such as wrong word choices and grammatical errors. In nearly all cases participants believed that prior knowledge of English influenced their opinions of the EBMT output. Interestingly, two participants pointed out that it was sometimes difficult to concentrate on the subtitles as they were also listening to the English audio track. This suggests to us that for our full-scale evaluation (which is outside the scope of this thesis) it would be a good idea to conduct some experiments where the audio track is muted, which should limit the interference of the source language.

Overall opinions

It is obvious from this preliminary study that our system is at too early a stage for it to be applied commercially. None of the participants would accept the raw EBMT-subtitles either on commercial DVD. However, all agreed that with some post-editing

of the output produced by the system would result in perfectly acceptable subtitles.

5.3 Discussion

In this chapter we have shown how various evaluation types have been used (1) during the development process to get the most out of our system and (2) as a means of determining whether output is of an acceptable level for public viewing. Through the use of automatic evaluation metrics we noticed that the nature of the corpus used to train the system has a huge impact on translation quality, and that our system performs best when seeded with homogeneous data, which meant that is was well worthwhile gathering together our own corpus of subtitles. We were also able to determine what chunk alignment strategies worked best for both language directions German–English and English–German, which was vital if we were to show our participants optimal EBMT-output in the pilot. One of the motivations for the project was to produce subtitles where they do not already exist; we were also able to demonstrate how our system is capable of translating bonus material to varying degrees of success. From our 'no context' human evaluation we were able to identify faults in our system; some of these aspects we were able to fix and some need to be improved upon in the future development of the system. Although the general consensus from our pilot study was that our subtitles are not yet at an acceptable viewing level, we believe we have created a solid framework for future studies into the acceptability of automatically generated subtitles.

Chapter 6

Conclusions and Future Work

In this thesis we have presented a comprehensive description of our work to date on using example-based machine translation in the production of foreign language subtitles. We began by outlining our aims and motivations for research, where we commented on the increasing amounts of pressure subtitlers and subtitle companies are being put under to produce high-quality foreign language subtitles for more and more language pairs in an ever-diminishing time frame. Through the use of a translation technology solution we have shown how EBMT could be used (1) similar to a translation memory tool to hopefully improve the throughput of the subtitler, and (2) to automatically produce foreign language subtitles in cases where they do not already exist.

We have described in detail our newly developed modular corpus-based MT system, MaTrEx, where each component was analysed individually and thoroughly. Although the corpora used to train the system were aligned at sentence level, the essence of EBMT is that it is able to identify subsentential examples and recombine these to produce a final translation. We have shown how the *marker hypothesis* can be easily implemented to chunk data into smaller linguistically intelligent units, and then presented how these chunks can be aligned thanks to a number of chunk similarity metrics such as: cognate information, part-of-speech tags, chunk length,

chunk position and word-to-word probabilities.

In order for us to determine what type of training corpus yielded the best results for our task of translating subtitles, we gathered together two types of corpora for training purposes: homogeneous data, which consisted of professionally translated subtitles, and heterogeneous data, which we acquired from European parliamentary proceedings. Extracting subtitles from DVD was a not a straightforward process as subtitles exist as bitmap images rather than raw text. Thus, we have given a thorough description of the subtitle extraction process and have shown how this data was aligned to give us our aligned bilingual corpus of subtitles. Subtitles can appear quite different to text from other domains, and we have noted how much shorter sentences from our subtitle data tend to be in comparison to our heterogeneous data. We performed some basic analysis on both corpora and then used the chi-square test to show how dissimilar our two corpora were to each other. In addition, we also explained how the chi-square test can be used to determine what words are *key* when comparing corpora to one another.

For our experiments we began by using automatic evaluation metrics common to MT to estimate the translation quality of our system output. Secondly, we designed and conducted two types of human evaluation, as we felt it extremely important to get a human's opinion on the subtitles produced by our system. During the developmental stage of our system, we noticed that the nature of the corpus used to train the system had a huge impact on translation quality. We discovered that our system performs consistently better when seeded with homogeneous data (previously translated subtitles) than when trained on equal or greater amounts of out-of-domain data. We were also able to determine which chunk-alignment strategies worked best for both language directions German-English and English–German, which was critical when it came to showing the participants optimal EBMT output in the pilot study. One of the aims of our study was to produce foreign language subtitles in

cases where they do not already exist, which we successfully demonstrated by using our system to translate DVD bonus material. From our 'harsh' type of human evaluation we were able to isolate certain faults in our system, some of which we were able to fix, and some which need to be addressed in the future development of the system. Although the results from our pilot study suggest that the output produced by our system is not yet at an acceptable level for public viewing, we believe we have created a solid framework for future studies into the acceptability of automatically generated subtitles. Finally, it is important to note that there are still a lot of improvements to be made to our EBMT system, and the general consensus from our pilot study was positive, with the vast majority of participants believing that once these changes are made there is a definite place for the use of EBMT in aiding the subtitler and also as a means of producing fully-automated subtitles in special cases where they do not already exist.

6.1 Future Work

As mentioned we were able to identify from our evaluation study certain errors made by our system. Word, chunk and sentence alignments form the basis of any EBMT system, and not all the alignments in our example database are perfect, which means that either we need to filter out these 'bad examples' or design more intelligent alignment techniques. During the recombination stage we make use of the statistical decoder Pharaoh (Koehn, 2004), which was designed for SMT systems. In order for us to get the most out of our example database, we would need to implement an example-based decoder (Groves, 2007). This has since been developed and has been shown to improve translation quality for other types of data; it would be interesting to apply this decoder for our task of translating subtitles to see how much of an impact it has on our system output.

For our study we concentrated on the language pair English–German. As our

system can easily be adapted to new language pairs we plan to see how well EBMT copes with these new languages, especially minority languages as our system could be used to provide people in these countries with subtitled audio-visual material, which they previously may not have had access to.

We have shown that the size and nature of the training corpus has a huge impact on translation quality. The problem remains that extracting subtitles, even via the advanced methods we used, is still a very time-consuming process. This is even more evident when extracting subtitles which contain characters not recognised by the OCR component of SubRip (Asian and Arabic character sets for example). Thus, for future studies involving a subtitle company or broadcasting authority to provide us with subtitles and their translations would save a lot of time in corpus creation.

As a final point I would like to mention that one of our aims to was use EBMT as a means of improving the throughput of the subtitler. Once improvements have been made to the system, we would like to integrate it into a subtitling suite, and test it out with some professional subtitlers to see if post-editing the EBMT-produced output results in a quicker translation turn-around time.

Bibliography

Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F., Purdy, D., Smith, N., and Yarowsky, D. (1999). Statistical Machine Translation. Technical Report for the John Hopkins University Workshop.

Armstrong, S., Caffrey, C., , Flanagan, M., O'Hagan, M., Kenny, D., and Way, A. (2006a). Improving the Quality of DVD Subtitles via Example-Based Machine Translation. In *Proceedings of the Translating and the Computer 28 Conference, [no page numbers]*, London, England.

Armstrong, S., Caffrey, C., and Flanagan, M. (2006b). Leading by Example: Automatic Translation of Subtitles Using EBMT. In *Presentation at the 6th International Conference and Exhibition on Language Transfer in Audiovisual Media. URL:* `http://www.computing.dcu.ie/~sarmstrong/LandM06.ppt` *[Accessed October 2006]*, Berlin, Germany.

Armstrong, S., Caffrey, C., and Flanagan, M. (2006c). Translating DVD Subtitles Using Example-Based Machine Translation. In *Presentation at the Multidimensional Translation Conference on Audiovisual Translation Scenarios Conference (MuTra-06). URL:* `http://www.computing.dcu.ie/~sarmstrong/mutra06.ppt` *[Accessed October 2006]*, Copenhagen, Denmark.

Armstrong, S., Flanagan, M., Graham, Y., Groves, D., Mellebeek, B., Morrissey, S., Stroppa, N., and Way, A. (2006d). MaTrEx: Machine Translation Using Examples. In *Presentation at TC-STAR OpenLab on Speech Translation. URL:*

http://www.computing.dcu.ie/~sarmstrong/openlab06.pdf *[Accessed October 2006]*, Trento, Italy.

Baayen, R. H., Piepenbrock, R., and Gulikers, L. (1995). The CELEX Lexical Database (CD-ROM).

Brown, P., Pietra, S. D., Pietra, V. D., and Mercer, R. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, 16:79–85.

Brown, P., Pietra, S. D., Pietra, V. D., and Mercer, R. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. In *Computational Linguistics*, pages 263–311.

Brown, R. (1999). Adding Linguistic Knowledge to a Lexical Example-based Translation System. In *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, pages 22–32, Chester, England.

Carroll, M. (2004). Subtitling: Changing Standards for New Media. *Web article:* http://www.translationdirectory.com/article422.htm *[Accesssed October 2006]*.

Cavaglia, G. (2002). Measuring corpus homogeneity using a range of measures for inter-document distance. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-02)*, pages 426–431, Las Palmas, Canary Islands.

Chandioux, J. (1976). METEO: un système opérationnel pour la traduction automatique des bulletins météorologiques. *Meta 21:127-133.*

Denoual, E. (2005). The Influence of Example-data Homogeneity on EBMT Quality. In *Proceedings of the Second Workshop on Example-Based Machine Translation, MT Summit X*, pages 35–42, Phuket, Thailand.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Journal of Ecology*, 26:297–302.

Doddington, G. (2002). Automatic evaluation of machine translation quality using N-gram co-occurrence statistics. In *Proceedings of Human Language Technology Conference (HLT-02)*, pages 138–145, San Diego, CA.

Gale, W. A. and Church, K. W. (1991). A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, pages 177–184, Berkeley, CA,.

Gambier, Y. (2005). Is Audiovisual Translation the Future of Translation Studies? In *Keynote Speech Delivered at the Between Text and Image, Screen-Translation Conference*, Forli, Italy.

Gough, N. (2005). *Example-Based Machine Translation using the Marker Hypothesis.* PhD thesis, Dublin City University, Dublin, Ireland.

Gough, N. and Way, A. (2004). Robust Large-Scale EBMT with Marker-Based Segmentation. In *Proceedings of the 10th Conference on Theoretical and Methodological Issues in Machine Translation (TMI-04)*, pages 95–104, Baltimore, MD.

Green, T. (1979). The Necessity of Syntax Markers. Two experiments with artificial languages. *Journal of Verbal Learning and Behaviour*, 18:481–486.

Groves, D. (2007). *Hybrid Data Driven Models of Machine Translation.* PhD thesis, Dublin City University, Dublin, Ireland.

Groves, D. and Way, A. (2005). Hybrid Example-Based SMT: the Best of Both Worlds? In *Proceedings of Association for Computational Linguistics 2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages 183–190, Ann Arbor, MI.

Groves, D. and Way, A. (2006). Hybridity in MT: Experiments on the Europarl Corpus. In *Proceedings of the 11th Conference of the European Association for Machine Translation*, pages 115–124, Oslo, Norway.

Hearne, M. and Way, A. (2003). Seeing the Wood for the Trees: Data-Oriented Translation. In *Proceedings of Machine Translation Summit IX*, pages 165–172, New Orleans, LO.

Hirschberg, D. (1977). Algorithms for the longest common subsequence problem. *Journal of the Association for Computing Machinery*, 24(4):664–675.

Hoey, M. (2006). Lexical priming: A new theory of words and language. *International Journal of Lexicography*, 19(3):327–335.

Jordan, P. W., Dorr, B. J., and Benoit, J. W. (1993). A First-Pass Approach for Evaluating Machine Translation Systems. *Machine Translation*, 8(1-2):49–58.

Joscelyne, A. (2006). Best Practices in Post-Editing. In *Translation Automation User Society (TAUS) Special Report (available to TUAS members only) [Accessed October 2006]*.

Jurafsky, D. and Martin, J. H. (2000a). HMMs and Speech Recognition. In *Speech and Language Processing*, page 156, Prentice Hall International (UK) Limited. London, UK.

Jurafsky, D. and Martin, J. H. (2000b). Probabilistic Models of Pronunciation and Spelling. In *Speech and Language Processing*, page 156, Prentice Hall International (UK) Limited. London, UK.

Kamprath, C., Adolphson, E., Mitamura, T., and Nyberg, E. (1998). Controlled Language for Multilingual Document Production: Experience with Caterpillar Technical English. In *Proceedings of the Second International Workshop on Controlled Language Applications: CLAW-98, [No page numbers]*, Pittsburgh, PA.

Kenny, D. (2006). Personal communication.

Kilgarriff, A. (2001). Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):1–37.

Koehn, P. (2003). Statistical Phrase-Based Translation. In *Proceedings of Human Language Technology Conference (HLT-NAACL)*, pages 48–54, Edmonton, Canada.

Koehn, P. (2004). Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *Proceedings of The 6th Conference of the Association for Machine Translation in the Americas, AMTA-04*, pages 115–124, Washington, DC.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the MT Summit X*, pages 79–86, Phuket, Thailand.

Lagoudaki, E. (2006). Tranlsation Memories Survey 2006: Users' perceptions around TM use. In *Proceedings of the Translating and the Computer 28 Conference, [no page numbers]*, London, England.

Leusch, G., Ueffing, N., and Ney, H. (2006). CDER: Efficient MT evaluation using block movements. In *Proceedings of 11th Conference of the European Chapter of the Association of Computational Linguistics, EACL-06*, pages 241–248, Trento, Italy.

Levenshtein, V. I. (1965). Binary codes capable of correcting spurious insertions and reversals. *Cybernetics and Control Theory*, 10:707–710.

Luyckx, K., Daelemans, W., and Vanhoutte, E. (2006). Stylogenetics: clustering-based stylistic analysis of literary corpora. In *Proceedings of the 5th International Language Resources and Evaluation Conference (LREC-06)*, pages 30–35, Genoa, Italy.

Mann, G. and Yarowsky, D. (2001). Multipath translation lexicon induction via bridge languages. In *Proceedings of the second meeting of the North American*

Chapter of the Association for Computational Linguistics, NAACL-01, pages 151–158, Pittsburgh, PA.

Muegge, U. (2006). Fully Automatic High Quality Machine Translation of Restricted Text: A Case Study. In *Proceedings of the Translating and the Computer 28 Conference, [no page numbers]*, London, England.

Nagao, M. (1984). A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In A. Elithorn and R. Banjeri (eds.) *Artificial and Human Intelligence*, pages 173–180, Amsterdam, North-Holland.

O'Brien, S. (2006). *Machine Translatability and Post-Editing Effort: An Empirical Study using Translog and Choice Network Analysis.* PhD thesis, Dublin City University, Dublin, Ireland.

Och, F. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of 41st Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

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

O'Hagan, M. (2003). Can Language Technology Respond to the Subtitler's Dilemma? - A preliminary study. In *Proceedings of Translating and the Computer 25, ASLIB, [no page numbers]*, London, England.

Papineni, K., Roukas, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 4th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Paris, C. (2002). ChapterXtractor 0.962, *URL:* `http://www.divx-digest.com/software/chapterxtractor.html` *[Accessed October 2006].*

Pigott, I. M. (1988). MT in Large Organizations: Systran at the Commission of European Communities. In M Vasconcellos (ed.) *American Translators Association Scholarly Monograph Series, Vol. II.*, pages 159–166.

Piperidis, S., Demiros, I., and Prokopidis, P. (2005). Infrastructure for a multilingual subtitle generation system. In *Proceedings of the 9th International Symposium on Social Communication*, Santiago de Cuba, Cuba.

Popowich, F., McFetridge, P., Turcato, D., and Toole, J. (2000). Machine translation of closed captions. *Machine Translation*, 15(4):311–341.

Povlsen, C. and Bech, A. (2001). Ape: Reducing the Monkey Business in Post-Editing by Automating the Task Intelligently. In *Proceedings of the MT-Summit VIII, Machine Translation in the Information Age*, pages 283–286, Santiago de Compostela, Spain.

Preece, J. (1993). A Guide to usability: human factors in computing. Technical report, Wokingham, UK.

Schäler, R., Carl, M., and Way, A. (2003). EBMT in a Controlled Environment. In M. Carl A. Way (eds.) *Recent Advances in Example-Based Machine Translation*, pages 83–114, K.A.P. Dordrecht, The Netherlands.

Scott, M. (2006). WordSmith Tools Manual Version 4.0. In *Technical manual for WordSmith Tools. URL:* `http://www.lexically.net/wordsmith/` *[Accessed October 2006].*

Simard, M., Foster, G. F., and Isabelle, P. (1992). Using cognates to align sentences in bilingual corpora. In *Proceedings of the 4th International Conference on Theoretical and Methodical Issues in Machine Translation (TMI-92)*, pages 1–11, Montreal, Canada.

Simard, M. and Langlais, P. (2001). Sub-sentential Exploitation of Translation Mem-

ories. In *Proceedings of MT Summit VII: Machine Translation in the Information Age*, pages 335–339, Santiago de Compostela, Spain.

Sinclair, J. (1991). *Corpus, concordance, collocation.* Oxford University Press.

Somers, H. (2003). An Overview of EBMT. In M. Carl A. Way (eds.) *Recent Advances in Example-Based Machine Translation*, pages 3–57, K.A.P. Dordrecht, The Netherlands.

Stolcke, A. (2002). SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP-2002)*, pages 1471–1474, Denver, CO.

Stroppa, N., Groves, D., Sarasola, K., and Way, A. (2006). Example-based Machine Translation of the Basque Language. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 232–241, Boston, MA.

Taylor, C. (2006). I knew he'd say that! A consideration of the predictability of language use in film. In *Presentation at the Marie Curie High Level Conference Series, Multidimensional Translation, Audiovisual Translation Scenarios*, Copenhagen, Denmark.

Thomas, J. (1996). *Using Corpora for Language Research.* Longman Publishing, New York.

Tiedemann, J. (2007). Improved Sentence Alignment for Building a Parallel Subtitle Corpus. In *CLIN 17 - Computational Linguistics in the Netherlands*, Leuven, Belgium.

Toda, N. (2005). Cited in The subtleties of subtitles. *Web article:* `http://www.crisscross.com/jp/newsmaker/266` *[Accessed October 2006].*

van Slype, G. (1980). Bewertung des Verfahrens SYSTRAN für die maschinelle Sprachbersetzung bei der K.E.G. *Lebende Sprachen: Zeitschrift für Fremde Sprachen in Wissenschaft und Praxis*, 25:6–9.

Wagner, E. (1985). Post-editing Systran - A Challenge for Commission Translators. In *Terminologie et Traduction, OPOCE, European Commission.*, London, England.

Wagner, R. and Fischer, M. (1974). The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1):168–173.

Watanabe, H., Kurohashi, S., and Aramaki, E. (2003). Finding translation patterns from paired source and target dependency structures. In M. Carl A. Way (eds.) *Recent Advances in Example-Based Machine Translation*, pages 397–420, K.A.P. Dordrecht, The Netherlands.

Way, A. and Gough, N. (2005). Comparing example-based and statistical machine translation. *Natural Language Engineering*, 11(3):295–309.

Zuggy (2006). Subrip 1.50 beta 4, *URL:* `http://zuggy.wz.cz/` *[Accessed October 2006].*