

Department of Computer and Information Science


Final Thesis

Evaluation of Two Word Alignment Systems

**By
Xiaoyang Wang**

LITH-IDA-EX--04/019--SE

Evaluation of Two Word Alignment Systems

 LINKÖPINGS UNIVERSITET	Avdelning, Institution Division, Department	Datum Date 2004-02-20
	Institutionen för datavetenskap 581 83 LINKÖPING	

Språk Language Svenska/Swedish X Engelska/English	Rapporttyp Report category Licentiatavhandling X Examensarbete C-uppsats D-uppsats Övrig rapport _____	ISBN	
		ISRN	LITH-IDA-EX--04/019--SE
		Serietitel och serienummer	ISSN
		Title of series, numbering	_____

URL för elektronisk version
<http://www.ep.liu.se/exjobb/ida/2004/dd-d/019/>

Titel Title	Evaluation of two word alignment systems
Författare Author	Xiaoyang Wang

Sammanfattning
 Abstract
 This project evaluates two different systems that generate word alignments on English-Swedish data. The systems to be used are the Giza++ system, that may generate a variety of statistical translation models, and I*Trix system developed at IDA/NLPLab that generates word pairs with frequencies.

The file formats of these two systems, the way of running them and the differences of the two systems are addressed in this paper. Evaluation in this project considers a variety of parameters such as corpus size, characteristics of the corpus, the effect of linguistic knowledge, etc. At the end of this paper, the conclusions of the two systems evaluation are presented. In general, Giza++ is better applying on big corpora while I*Trix is better for small corpora. Especially for corpora with high statistical ratio or special resource, I*Trix has a better performance.

Nyckelord
 Keyword
 Word alignment, Giza++, I*Trix, Parallel corpora, Statistical ratio, Evaluation, I*Eval, Gold standard.

Evaluation of Two Word Alignment Systems

Evaluation of Two Word Alignment Systems

Degree report
by
Xiaoyang Wang

ISRN: LITH-IDA-EX--04/019--SE
Feb 2004

Supervisor: Lars Ahrenberg
Examiner: Lars Ahrenberg

ABSTRACT

This project evaluates two different systems that generate word alignments on English-Swedish data. The systems to be used are the Giza++ system, that may generate a variety of statistical translation models, and I*Trix system developed at IDA/NLPLab that generates word pairs with frequencies.

The file formats of these two systems, the way of running them and the differences of the two systems are addressed in this paper. Evaluation in this project considers a variety of parameters such as corpus size, characteristics of the corpus, the effect of linguistic knowledge, etc. At the end of this paper, the conclusions of the two systems evaluation are presented. In general, Giza++ is better applying on big corpora while I*Trix is better for small corpora. Especially for corpora with high statistical ratio or special resource, I*Trix has a better performance.

KEYWORDS

Word alignment, Giza++, I*Trix, Parallel corpora, Statistical ratio, Evaluation, I*Eval, Gold standard.

Evaluation of Two Word Alignment Systems

ACKNOWLEDGEMENTS

As the mark of the end of my Master Program study in Linköpings Universitet, this thesis report means a lot to me.

First of all, I would like to express my thanks to my examiner and supervisor, Lars Ahrenberg, who supported the studies and supervised me with a lot of patience. He has guided me in the work that has resulted in this thesis. Despite a very busy schedule he has always been available for questions and discussions, and he has read and commented on a great number of my drafts of this paper.

Many thanks also goes to Michael Petterstedt, who implemented some of the tools I used in this project. He explained to me a lot about the systems and gave me some important advices. Without him, this thesis work could not have been accomplished.

I also would like to thank Jin Nie, who worked in the same room with me. We worked for different parts of the same project and he gave me many good suggestions during my work and taught me some good habits of work.

When I have had questions regarding technical parts or programming, Xiaogang Zhang has helped me a lot, many thanks for this. Thanks also to other people who have ever helped me or provided such pleasant atmosphere.

Evaluation of Two Word Alignment Systems

CONTENTS

ABSTRACT

KEYWORDS

ACKNOWLEDGEMENTS

CONTENTS

1. Introduction	1
1.1 Goal and scope of project	1
1.2 Overview of report	1
2. Background.....	3
2.1 Parallel corpora	3
2.2 Sentence alignment.....	4
2.3 Word alignment.....	4
2.3.1 Association approaches	6
2.3.2 Estimation approaches	6
2.4 Evaluation	9
2.4.1 Measuring Methods	10
2.4.2 Gold standard.....	10
2.5 Summary	12
3. The main systems used in this project	13
3.1 Giza++ system.....	13
3.1.1 Running Giza++ in IDA.....	13
3.1.2 Input file formats in Giza++	13
3.1.3 Output file formats in Giza++	14
3.2 I*Trix system	16
3.2.1 Interface of I*Trix.....	16
3.2.2 Input file formats in I*Trix.....	19
3.2.3 Output file formats in I*Trix	19
3.3 Evaluation tool – I*Eval	20
4. The evaluation environment.....	25
4.1 Symmetrization	25
4.1.1 Inputs symmetrization	25
4.1.2 Outputs symmetrization	26
4.1.3 Resources used.....	26
4.2 Participating Corpora.....	27
4.2.1 Blocks Corpus.....	27
4.2.2 Access XP 97 sentences Corpus	28
4.2.3 Access XP 5000 sentences Corpus	28
4.2.4 Corpora Summary	28
5. Analysis results	29
5.1 Things that may influence the results	29
5.1.1 In Giza++ system	29
5.1.2 In I*Trix system.....	30
5.1.3 In I*Eval.....	30
5.2 Comparing the results from the two systems.....	30
5.2.1 Parameter Setting in I*Trix	30
5.2.2 Word classes in Giza++	36
5.2.3 Different sizes of corpora	37

Evaluation of Two Word Alignment Systems

5.2.4 Repeated corpora	39
5.2.5 Monolingual corpora.....	41
5.3 Speed of two systems	42
6. Summary and Conclusions.....	43
6.1 Summary	43
6.2 Evaluation results related to parameter setting	43
6.3 Evaluation results related to corpora	44
6.4 Strengths and weaknesses.....	44
6.5 Future work.....	45
7. References	47
Appendix A: Concept definitions.....	51
Appendix B: Running Giza++ at IDA.....	53
Appendix C: Code for xml2text	55

1. Introduction

In recent years more attention has been paid to the fields of translation studies and corpus-based machine translation with Statistical Machine Translation. The basic idea is to use the data in one language and its pair translation data in other language together to automatically train a translation model and a language model that can be used for developing a decoder that performs translation.

1.1 Goal and scope of project

This project evaluated two different systems that generate word alignments on English-Swedish data. The systems to be used are, on the one hand, the Giza++ system that generates a variety of statistical translation models, on the other hand, the I*Trix system developed at IDA/NLPLab that generates word alignment pairs with frequencies. The results of this project could be used for future study at IDA/NLPLab.

Evaluation is not only to compare the strengths and weaknesses of the two systems, but also to know what to improve in a system for the system developers and to help the user understand a system better. Evaluation should be systematic and consider a variety of parameters. In this project, attention has been paid to the characteristics of the corpora such as size and type/token-ratio, and, for the I*Trix system, selection and weighing of linguistic data.

1.2 Overview of report

The rest of this report deals with the following subjects:

Section 2: A background to the relevant areas within parallel corpora, word alignment and evaluation is presented, which will set the ground for later discussion about Giza++ system, I*Trix system and the evaluation environment and methods.

Section 3: The details of the used tools are studied in this section. It includes the inputs and outputs of the tools, how to make them running and so on. The tools used in this project are Giza++, I*Trix and I*Eval.

Section 4: Before the evaluation, a lot of things need to be done to make the evaluation possible. Implementation of an environment for the performance of the evaluation is discussed in this section.

Section 5: Some tests are applied to evaluate different results from two different systems. First, very detailed studies about the parameter setting in I*Trix are presented. Some other tests on different corpora are made to analyse the results. At the end, the speeds of two systems are compared.

Section 6: Conclusions, discussion and future work.

Evaluation of Two Word Alignment Systems

2. Background

In this chapter, basic knowledge in the field of translation corpus processing, which is related to the two evaluated systems Giza++ and I*Trix, is introduced. Also, the background about evaluation is mentioned to make clear what the main purpose of this project is.

2.1 Parallel corpora

In 1961, the work on the text corpus, which is later known as the Brown Corpus (Francis and Kucera 1964), was first started. That is the advent of corpus linguistics. The use of “corpus linguistics” has however become associated with language material that exists in electronic formats and the various methods and software tools that are used to analyse and access such data. The increasing processing power and storage capacity of computers have not only meant that the number of available text corpora has increased, but also that text corpora are larger in size, more varied and easier to access.

The original meaning of corpus in Mona Baker’s words was “any collection of writings, in a processed or unprocessed form” (Baker 1995). But now, the definition has to be modified because of the growth of corpus linguistics. “(i) Corpus now means a collection of texts held in machine-readable format and capable of being analyzed automatically or semi-automatically in a variety of ways; (ii) a corpus is no longer restricted to ‘writings’ but includes spoken as well as written text, and (iii) a corpus may include a large number of texts from a variety of sources” (Baker *ibid*). However, a corpus is always just a sample and can never completely represent a whole language. The expressive power of natural language cannot be captured by a finite data set.

Depending on the texts properties, corpora can be generally divided into different types. Concerning whether the corpus contains texts in one or more than one languages, there are monolingual corpora and multilingual corpora (Merkel 1999). Multilingual corpora can be divided into parallel corpora and non-parallel corpora (Merkel *ibid*).

Parallel corpora are referred to natural language utterances and their translations with alignments between corresponding segments in different languages. Parallel corpora usually contain a common source document and one or more target documents. Bilingual parallel corpora are sometimes called **bitexts** (Isabelle 1992) and corresponding parts within these corpora are called bitext segments (Ahrenberg et al. 1999).

There are many applications using parallel corpora for translation studies and for tasks in multilingual natural language processing (NLP). Parallel corpora have become more widely available and serve as a source of NLP tasks in recent years. Statistical machine translation is just a research fields that has been developed in connection with a growing number of large parallel corpora.

Most parallel corpora contain only two languages, a source and a target language. A source and target language sentence that are translations of each other are called **sentence pair**. In this paper, talking about parallel corpora, English is used as source language while Swedish is target language. However, multilingual parallel corpora with translations into more than one language are available and became popular in recent studies.

2.2 Sentence alignment

Based on different purposes, source language text can be split into different segments that correspond monotonically to segments in the target text. Common segmentation units are paragraphs and sentences.

Alignment is defined as an object for indicating the corresponding words and phrases in a parallel text (Brown et al. 1993). In simple words, alignment tells which units in **target text** links to the units in **source text**. Alignment corresponding sentences is called sentence alignment (Merkel 1999). The sentence terminators are full stop, question mark and exclamation mark. In sentences alignment, some groups of sentences in one language correspond in content to some groups of sentences in the other language. Such an alignment essentially creates partly searchable parallel corpora of source and target text. It usually assumes that information at the sentence level is expressed in the same order in the original text as in its target text. With this assumption, sentence alignment can be regarded as an alignment without crossing **links**.

There are two main approaches to sentence alignment, named length-based alignment and lexicon-based alignment (Tiedemann 2003). The length-based alignment approach can be based on either character length (Gale and Church 1991) or word length (Brown et al. 1991). The lexicon-based alignment approach is based on words and other lexical units, or combinations of both.

2.3 Word alignment

Word alignment is a process of determining which words in a given source sentence should be translated to the words in a given target sentence. Word alignment is the basic task in extracting bilingual lexicons.

A simple word alignments example is shown graphically in Figure 2.1. The lines in the figure are called connections (Brown et al. 1993), from some of the English words to some of the Swedish words. For one pair of sentences, there are always more than one way of all the words connections. One way of words connections assignment is an alignment. Sometimes, words in the source sentence of the pair align with nothing in the target sentence, and similarly, words in target sentence may link to none of the words in the source sentence.

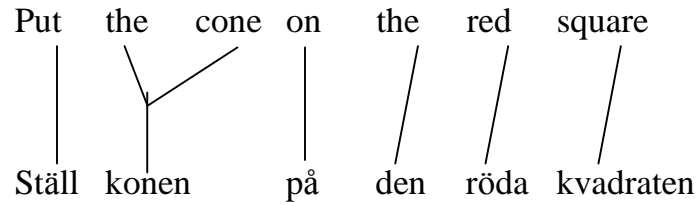


Figure 2.1 A word alignment example

The alignments between two sentence pairs can be very complicated. Even for a human being, judging which words in given target strings correspond to which words in its source strings is sometimes a tricky problem. The main reason is that the notion of “correspondence” between words is subjective.

One main purpose of word alignment is to produce lexical data for bilingual dictionaries, while another purpose can be to provide data for machine translation or contrastive studies. Notable, in a pure word-to-word model (cf. Melamed 1995), many valid lexical units are missed because of the fact that they belong to collocations or complex paraphrases. Word alignment system, which is based on the knowledge of word alignment, refers to systems that align linguistic units below the sentence level across two languages. The task of this kind of system is to find all correspondences at the lexical level that exists in a given parallel text. For all kinds of word alignment linking, to be able to handle multi-word segments in both the source and target text is necessary. In previous works, there are some approaches used for word alignment systems to handle multi-word. Both the Giza++ system and the I*Trix system are this kind of word alignment systems.

There are many translational relations between words and phrases in parallel corpora. Word order is in general not identical for most language pairs, and boundaries of lexical units are not as easy to detect as sentence boundaries. There is no consistent correlation between the character lengths of corresponding words, at least not for most language pairs. However, natural languages must be compositional in some sense for translation to be possible at all. Thus, many translation relations between these compositional components, i.e. words and phrases, can be found in documents and their translations.

The type of relation between words varies in parallel texts while the strategy of aligning words and phrases depends on the task. Usually, word alignment aims at a complete alignment of all lexical items in the corpus. The degree of correspondence can be expressed in terms of alignment probabilities, which is useful for statistical machine translation.

There are two approaches to word alignment. The association approach uses measures of correspondence of some kind. It also refers to as heuristic approaches (Och and Ney 2003) or hypothesis testing approaches (Hiemstra 2003). The estimation approach uses probabilistic translation models. It is often called statistical alignment (Och and Ney 2000a). Both approaches use some kind of statistics. These two approaches are briefly introduced and discussed in the following part.

2.3.1 Association approaches

Association approaches to word alignment originate mainly from early studies on lexical analysis of parallel data. In a dictionary extraction task it is not necessary to align all occurrences of a lexical item to corresponding items in the translation. In general the steps which have to be taken for aligning bitexts using an association approach are (Tiedemann 2003):

1. **Lexical segmentation:** Boundaries of lexical items have to be identified for both languages.
2. **Correspondence:** Possible translation relations between lexical items have to be identified according to some correspondence criteria. This usually results in a collection of weighted word type links, i.e. a translation dictionary with association scores attached to its entries. Contextual features may be attached to this association dictionary.
3. **Alignment and extraction:** The most reliable translations according to the association dictionary are marked in the bitext (alignment). This is commonly done in a “greedy” way using simple search strategies such as a “best first” search in combination with some linguistic/heuristic constraints. A bilingual translation dictionary can be compiled from the aligned items (extraction), which is usually “cleaner” than the previously produced association dictionary.

In association approach, co-occurrence measures and string similarity measures are two main alignment techniques (Tiedemann 2003).

The one-to-one alignment assumption for word alignment is insufficient for most language pairs. In word alignment, several studies have been made on the integration of **multi-word units (MWUs)**, which refer to word sequences and word groups (cf. Tiedemann 2003). Dealing with MWUs, there are two general techniques: prior identification of collocations and dynamic construction of MWUs during the alignment process.

The use of structural information in bilingual term extraction has been investigated in many studies, for example in Van der Eijk (1993). There are also some studies using statistical or hybrid approaches for the identification of phrasal units in bitext segments before proceeding with word alignment, see Ahrenberg et al (1998).

2.3.2 Estimation approaches

Estimation approaches are referred to word alignment using probabilistic alignment models that are estimated from parallel corpora (Tiedemann 2003). Most work in this field has been inspired by the work on statistical machine translation (SMT) introduced in Brown et al. (1990). Principles of statistical machine translation, the application of such models to word alignment and lexicon extraction tasks are reviewed briefly here.

The purpose of SMT is, based on the knowledge got from the known bilingual corpus, given the source sentence in one language, to find the most probable translation in another language. The difference between traditional machine and statistical machine translation is that, traditional machine translation is rule-based, while statistical translation is data driven.

The source language S and the target language T are considered to be random variables that produce strings such as sentences. Given a target language string t , the source language string s from which the translator produced t is looked for. Choosing the sentence s that is most probable given t minimizes the chance of error. It means that an s is chosen so as to maximize $Pr(s/t)$.

From Bayes' theory, equation (2.1) can be written:

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

The denominator $P(t)$ does not depend on s , so it suffices to choose the s that maximizes the product $P(s)P(t/s)$. The first factor in this product is called the language model probability of s and the second factor is the translation probability of t given s .

According to the model, the most likely solution can be found by using the $argmax_s$ function, which returns the argument s out of all possible values for s that maximizes the given function:

$$s = arg \max_s P(s | t) = arg \max_s \frac{P(t | s)P(s)}{P(t)} \quad (2.2)$$

Due to the fact that $P(t)$ is independent of s and, consequently, is constant for all possible strings s ; it can be ignored in the maximization procedure. The fundamental equation of the SMT model is therefore expressed as the following search problem:

$$s = arg \max_s P(t | s)P(s) \quad (2.3)$$

The translation model $P(t/s)$ is to be estimated from sentence-aligned parallel corpora. However, estimating $P(t/s)$ directly from a corpus is impossible because of the sparse data problem. The majority of segments in any parallel corpus, let it be as big as possible, will be unique. Even worse, most of the possible sentences s and t of two languages will not occur in any training corpus and therefore according parameters for a translation model are impossible to estimate. Consequently, the translation model has to be decomposed into distributions of smaller units, which recur more frequently in the training data and are more likely to appear again in unseen data. The first step in decomposing the general translation model is to introduce another random variable A denoting the alignment between sub-strings (i.e. words) of the source and the target language strings. Using all possible alignments a between s and t , the translation model can be re-written as follows:

$$P(t | s) = \sum_a P(t, a | s) \quad (2.4)$$

The alignment in SMT is usually modelled as a sequence of hidden connections between words in the target language string and words in the source language string. Specifically, each word in the target language string t is connected to exactly one word in the source language string s , which can be expressed as natural number representing the position of the connected source language word in the sentence.

Most translation models that have been used in the SMT community are based on five models introduced in Brown et al. (1993) by researchers at IBM. The idea behind these five-models is to start with a very simple model before progressing to more complex ones. The output of simpler models can be used in this way to initialize the following models.

Model 1 is a simplest word translation model. In Model 1 and 2, first choose a length for the source string, assuming all reasonable lengths to be equally likely. Then for each position in the source string, how to connect it to the target string and what source word to place need to be decided. In Model 1 all alignments have the same probability. Model 2 uses a zero-order alignment model where different alignment positions are independent from each other. Although it is possible to obtain interesting correlations between some pairs of frequent words in the two languages using Models 1 and 2, these models often lead to unsatisfactory alignments.

In Models 3, 4, and 5, from Brown et al.(1993), it is the last step that determines the connections between the source strings and target strings in these three models differ. In Model 3, the probability of a connection depends on the positions that it connects and on the lengths of the source and target strings. In Model 4 the probability of a connection depends in addition on the identities of the source and target strings connected and on the positions of any other source strings that are connected to the same target strings. Model 5 is similar to Model 4 except that it is not deficient.

Many works have been done in the area of machine translation. One of famous project is **EGYPT** (Knight et al. 1999), the toolkit developed by Statistical Machine Translation team during the summer workshop in 1999 at the Centre for Language and Speech Processing at Johns-Hopkins University (CLSP/JHU). It is built up from four tools:

- | **Whittle** is a tool for preparing and splitting bilingual corpora into training and testing sets. Whittle generates *.snt file formats (the input corpus format required by GIZA). Whittle is written in Perl.
- | **Giza** is a training program that learns statistical translation models from bilingual corpora. GIZA is written in C++ with the STL library (tested using gnu C++).
- | **Cairo** is a word alignment visualization tool written in Java.
- | **Cairoize** is a tool for generating alignments files in *.aln format (the format required by cairo). Cairoize is written in C and perl.

In this paper, one of the evaluation systems is Giza++. **Giza++** is the extension of GIZA. It was designed and written by Franz Josef Och (Och 2003). Some translation models introduced by IBM scientists in early 1990's are used in Giza++. The program includes the following extensions, compared to GIZA:

- | Model 4;
- | Model 5;
- | Alignment models depending on word classes (software for producing word classes are also developed);
- | Implements the HMM alignment model: Baum-Welch training, Forward-Backward algorithm, empty word, dependency on word classes, transfer to fertility models.
- | Includes a variant of Model 3 and Model 4 which allow the training of the parameter p_0 ;
- | Various smoothing techniques for fertility, distortion/alignment parameters;
- | Significant more efficient training of the fertility models;
- | Correct implementation of pegging as described in (Brown et al. 1993), a series of **heuristics** in order to make pegging sufficiently efficient.

2.4 Evaluation

Evaluation is a way for the system developers to know where and what to improve in a system and for the users to be aware of strengths and weaknesses of the systems. The main purpose of this project is to evaluate Giza++ system and I*Trix system. It is not a simple thing just to tell which system is good. There are many aspects is worthy to be considered in this evaluation.

The alignment between source and target sentences can be quite complicated. Especially, it is sometimes difficult for a human to judge which words in a given target string correspond to which words in its source string. But in order to evaluate two word alignment systems, human's opinion needs to play an important role.

If just using the evaluation tool to compare the alignment results from two systems, in one point of view, it can be thought meaningless. Because the links in the two systems might be both correct or both wrong. The results from the evaluation tool can only tell how the two results differ from each other. There is no standard to tell which system is better.

The most straightforward way to evaluate in this project is to compare the alignment results from two word alignment systems with human's opinion separately to see which of them is closer to the gold standard.

In addition, the running time and the costs of two systems can be compared. Which system is easier to run and what is the easiest size of corpus for the systems to handle are also interesting aspects to be evaluated.

2.4.1 Measuring Methods

Evaluation Method should be tailored to a specific type of alignment system in order to avoid unfair comparison. Evaluations were performed with respect to four different measuring methods in this project: precision, recall, AER and F-measure. Among them, precision, recall and F-measure represent traditional measures in Information Retrieval and were frequently used in previous word alignment literature. Another method, AER, was originally introduced by (Och and Ney 2000a), and proposed the notion of quality of word alignment.

Given an alignment A of a bitext and a gold standard alignment (reference alignment) Ar . The recall, precision, F-measure and AER of the alignment A with respect to the gold standard Ar can be defined respectively as:

$$recall = \frac{|A \cap Ar|}{|Ar|} \quad (2.5)$$

$$precision = \frac{|A \cap Ar|}{|A|} \quad (2.6)$$

$$AER = 1 - \frac{recall * |Ar| + precision * |A|}{|Ar| + |A|} = 1 - \frac{|A \cap Ar|}{|Ar|} \quad (2.7)$$

$$F - measure = \frac{2 * precision * recall}{precision + recall} = \frac{2 * |A \cap Ar|}{|A| + |Ar|} \quad (2.8)$$

The recall measurement gives the proportion of correct segments (in gold standard) that is in proposed alignment A . The precision measurement gives the proportion of segments in the proposed alignment A that is considered to be correct. These two measurements are straightforward to handle if the text only consists of single words, but it becomes increasingly difficult when the alignments are multiple.

AER is the Alignment Error Rate with respect to the gold standard Ar . The smaller number in AER is the closer A and Ar is. (See equation 2.7).

F-measure is a straightforward measurement to tell how the proposed alignment similar with the gold standard. The range of F-measure is from 0 to 1. If all the links in A and Ar are same, F-measure is 1. If no links in A and Ar are the same, then F-measure is 0. (See equation 2.8)

2.4.2 Gold standard

Gold standard is usually a sample of the bitext that has been pre-aligned manually by one or several **annotators** and then used to test the alignment output automatically. Gold standard plays an important role in evaluation. Doing the evaluation in this project the output of the two systems will be compared with gold standard. But usually, it's very difficult to get gold standard files because they are manually linked. Especially for big corpora, it's almost impossible to get all the sentences manually linked until now. This is a common main problem for most of the evaluations of word alignment systems. Usually, the way to deal with this problem is to use the gold

standard with aligning a subset of sentences in the whole corpus, not all the sentences pairs.

In some studies sample bitext segments have been completely aligned by hand in order to create gold standard (Melamed 1998), (Och and Ney 2000b). In other studies, like Ahrenberg et al. (2000) and Véronis and Langlais (2000), word samples from the corpus were used.

In this project, all the gold standard files are in link file format, which will be explained in detail in 3.2.3, and the annotators created them by using I*Link (Ahrenberg 2000). In this kind of gold standard, the words both in source and target sentences can be linked to null respectively. One word can also be linked to multiple words. Some words might be linked to nothing depends on the annotators.

Link strategy is worthy to be mentioned when gold standard is introduced. Link strategy is a group of rules for the annotators to learn before they start to link. Link strategy comes from the purpose of the alignment. Different purposes might need different gold standard. Link strategy influences two things related to this project.

- Null link split
If multiple continuous words need to be linked to null, they might be either linked to multiple null links or to one null link. For example:

Source sentence:

This makes it difficult to allow other users to gain access to the Access project.

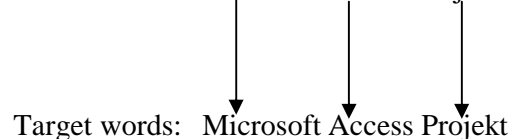
Target sentence:

Detta gör det svårt att ge andra användare tillträde till Access-projektet.

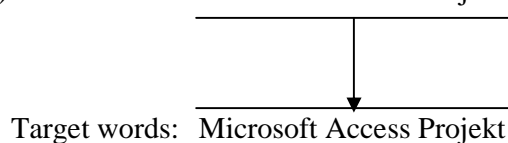
In the gold standard, “to gain” can be linked to “NULL NULL” or only to “NULL”. In this project, all the gold standard alignments will split the null links, which means the group of words will be linked to multiple null links if meet this kind of situation.

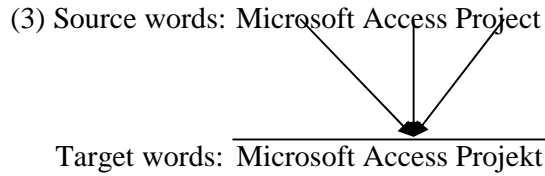
- Multiple word units (MWUs)
Based on different link strategy, multiple words link to multiple words might have several ways. For example, all the following ways of links are possible:

(1) Source words: Microsoft Access Project



(2) Source words: Microsoft Access Project





All the gold standard alignments in this project follow the second way of linking multiple words. Maximum multiple words in source sentences will try to link to maximum multiple words in target sentences.

2.5 Summary

In this part, basic concepts and techniques of the work with parallel corpora have been presented. Common methods for the alignment of sentences have been discussed. Two main approaches to word alignment have been mentioned, which are association approaches and estimation approaches. Association measures are widely used for the identification of translational correspondences. In statistical machine translation, probabilistic alignment models are used to estimate alignment parameters between words in parallel corpora. In this paper the approach is referred to as the estimation approach to word alignment. The purpose and methods of evaluation are proposed in this chapter. A brief introduction of gold standard is described at the end of this chapter.

3. The main systems used in this project

3.1 Giza++ system

GIZA++ is an extension of the program GIZA (part of the SMT toolkit EGYPT) which was developed by the Statistical Machine Translation team during the summer workshop in 1999 at the Centre for Language and Speech Processing at Johns-Hopkins University (CLSP/JHU). It is a program for learning statistical translation models from bitext. GIZA++ includes a lot of additional features. The extensions of GIZA++ were designed and written by Franz Josef Och (Och 2003).

3.1.1 Running Giza++ in IDA

Giza++ is a program that has to follow a serial of command steps to run. Different environments might need different steps and commands. In Appendix B, there are the steps needed to do in IDA Solaris machine to run Giza++.

Noticeable, running “mkcls” is an optional step. “mkcls” is a tool to train word classes by using a Maximum-likelihood-criterion. The resulting word classes are especially suited for language models or statistical translation models. The program “mkcls” was written by Franz Josef Och (Och 2001).

3.1.2 Input file formats in Giza++

After the first important step which is running the program “plain2snt.out”, “*.vcb” files and “*.snt” file will be created. “plain2snt.out” is a simple tool to transform plain text into GIZA++ text format. These three files are input files for basic Giza++ running.

VCB is abbreviating of Vocabulary file. Each entry is stored in one line as follows:

```
uniq_id1 string1 no_occurrences1
uniq_id2 string2 no_occurrences2
uniq_id3 string3 no_occurrences3
....
```

SNT is a Bitext file. Each sentence pair is stored in three lines. For example an English-Swedish sentence pair is:

*Each category consists of one point from each data series .
Varje kategori består av en poäng från varje dataserie .*

The SNT file for this sentence pair is:

```
1
2 3 4 5 6 7 8 9 10 11 12
2 3 4 5 6 7 8 9 10 11
```

The first line is the number of times that this sentence pair occurred. The second line is the source sentence where each token is replaced by its unique integer id from the vocabulary file and the third line is the target sentence in the same format.

After running “mkcls”, four output files are created. In the “*.vcb.classes” file, all the unique words in the whole text and the numbers of times the words appear are listed. The files “*.vcb.classes.cats” indicates which class every word belonged to.

3.1.3 Output file formats in Giza++

Giza++ has 17 output files after running. In this project, only the A3.final file is used as Giza++ output alignment result. The way of evaluation is to convert this file format to link file format, which will be discussed later. So except the A3.final file, only a very brief introduction about part of important output file formats is mentioned here to give an idea what the output results of Giza++ are.

- a3.final
The format in a3 file of each line is as follows:
i j l m p(i | j, l, m)
where i, j, l, m are all integers and
j = position in target sentence
i = position in source sentence
l = length of source sentence
m = length of target sentence
p(i/j,l,m) is the probability that a source word in position i is moved to position j in a pair of sentences of length l and m.
- t3.final
Each line is of the following format: s_id t_id P(t_id/s_id), where:
s_id: is the unique id for the source token
t_id: is the unique id for the target token
P(t_id/s_id) the probability of translating s_id as t_id
Similar files will be generated (with the prefix "prob_table.actual.xxx" that has the actual tokens instead of their unique ids). This is also true for fertility tables. Also the inverse probability table will be generated for the final table and it will have the infix "ti".
- Revised Vocabulary files (*.src.vcb, *.trg.vcb)
The revised vocabulary files are similar in format to the original vocabulary files (VCB files). The only exception is that the frequency for each token is calculated from the given corpus, which is not required in the input.

In “*.src.vcb” file, each line consists of information of source corpus:
Vocabulary id
Vocabulary
The number of times that word appears in the corpus
In the “*.trg.vcb” file, each line consists of similar information of target corpus in target language.

- **d4.final**
This file contains position information for headword and body words. For head position, it comes like that:
Source word class
Target word class
Position
Probability
For body word, it comes out like that:
Target word class
Position
Probability
- **ti.final**
This is the translation probability, but in inverse direction, each line contains:
Target word id
Source word id
Translation probability
- **n3.final**
This is fertility table. Each line in this file is of the following format:
Source_token_id p0 p1 p2.... pn
p0 is the probability that the source token has zero fertility. p1, fertility one, and n is the maximum possible fertility as defined in the program.
- **P0-3.final**
This file contains only one line with one real number that is the value of P0, the probability of not inserting a NULL token.
- ***.gizacfg**

This file includes all the parameter settings that are used in order to perform this training. This means that starting GIZA using this parameter file produces (should produce) the same training.
- **Perplexity File (*.perp)**

This file will be generated at the end of the training. It summarizes perplexity values for each training iteration. The format is the same for cross entropy. If no test corpus is provided, the values for it will be set to "N/A".
- ***.A3.final**

In each process of training iteration, and for each sentence pair in the training set, the best alignment (Viterbi alignment) is written to the alignment file (if the dump parameters are set accordingly). The alignment file is named prob_table.An.i, where n is the model number ({1,2, 2to3, 3 or 4}), and i is the iteration number. The format of the alignments file is illustrated in the following sample:

```
# Sentence pair (1) source length 11 target length 10 alignment score: 9.06432e-06
Varje kategori består av en poäng från varje dataserie.
NULL ( { } ) Each ( { 1 } ) category ( { 2 } ) consists ( { 3 } ) of ( { 4 } ) one ( { 5 } )
point ( { 6 } ) from ( { 7 } ) each ( { 8 } ) data ( { } ) series ( { 9 } ) . ( { 10 } )
# Sentence pair (2) source length 26 target length 20 alignment score: 6.69883e-14
Kategorietiketter visas oftast över diagrammets x-axel, vilket dock kan variera
beroende-på vilken typ av diagram som du använder.
NULL ( { 17 } ) Category ( { 1 } ) labels ( { 2 } ) usually ( { 3 } ) appear ( { 4 } ) across
( { 5 } ) the ( { } ) x ( { 6 } ) axis ( { 13 14 } ) of ( { } ) the ( { } ) chart ( { } ) , ( { 7 } )
although ( { 8 } ) this ( { } ) can ( { 10 } ) vary ( { 9 11 } ) depending ( { 12 } ) on ( { } )
the ( { } ) type ( { 19 } ) of ( { 15 } ) chart ( { 16 } ) you ( { 18 } ) are ( { } ) using ( { } ) .
( { 20 } )
```

Three lines for each sentence pair represent the alignment file. The first line is a label that can be used, e.g. as a caption for alignment visualization tools. It contains information about the sentence sequential number in the training corpus, sentence lengths, and alignment probability. The second line is the target sentence, which is Swedish in this example. The third line is the source sentence. Each token in the source sentence is followed by a set of zero or more numbers. These numbers represent the positions of the target words to which this source word is connected, according to the alignment.

3.2 I*Trix system

I*Trix is an automatic tool for creating and storing associations between segments in a bitext. I*Trix is aimed at word and phrase associations and requires bitexts that are pre-aligned at the sentence level. An important observation is that bitext linking is the process of creating links between textual units or segments of a source and a target text for a certain purpose.

I*Trix is developed at the Natural Language Processing Laboratory (NLPLAB), Department of Computer and Information Science at Linköping University, Sweden, with funding from The Swedish Research Council (Vetenskapsrådet) and The Swedish Agency for Innovation Systems (VINNOVA). I*Trix is developed by Michael Petterstedt as a general tool for creating and classifying word associations in parallel corpora, and should be useful for different kinds of research including translation studies, contrastive linguistics and machine translation.

I*Trix was first thought of as a plug in module for I*Link, e.g. the main heuristic module for the alignment process. During the development of this module, however, it turned out that this automatic alignment tool could be quite powerful in its own perspective.

3.2.1 Interface of I*Trix

I*Trix is a program with a friendly interface. Compared with Giza++, making I*Trix run is very easy. Figure 3.1 is the interface of I*Trix. It includes bitexts segment area, alignment area and function area.

3. Main tools related in this project

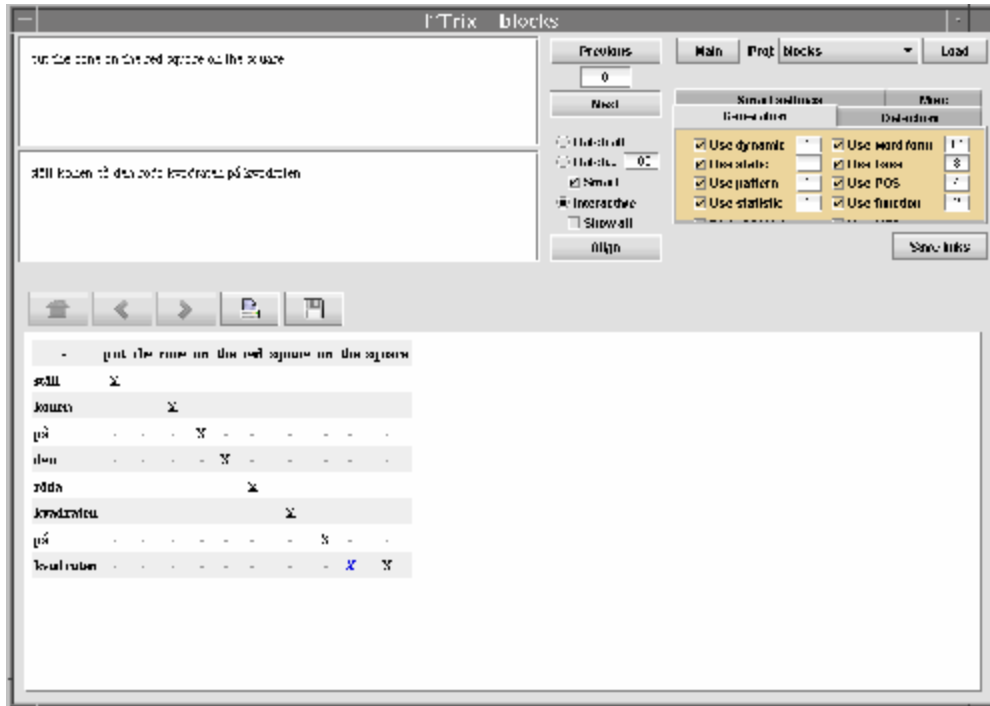


Figure 3.1 Interface of I*TriX

1. **Bitexts Segment Area** presents a current sentence pair of the source and target texts. (Figure 3.2)

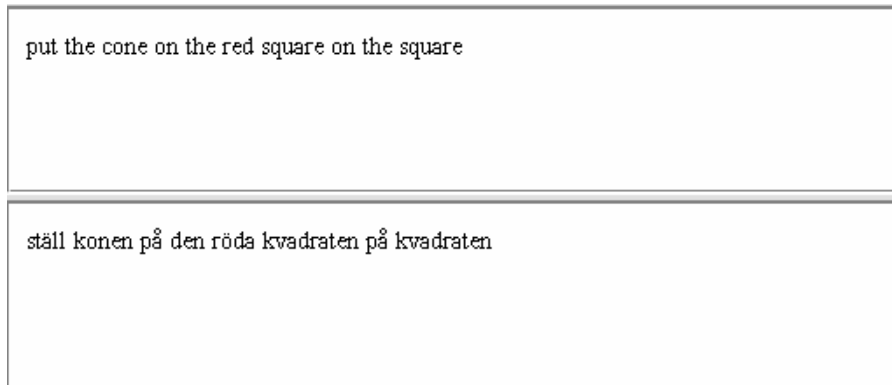


Figure 3.2 Bitext Segment Area

2. **Alignment Area** shows the alignment result for the current sentence pair after applying the current parameter settings. See Figure 3.3. This result can be saved or printed out by the function buttons.

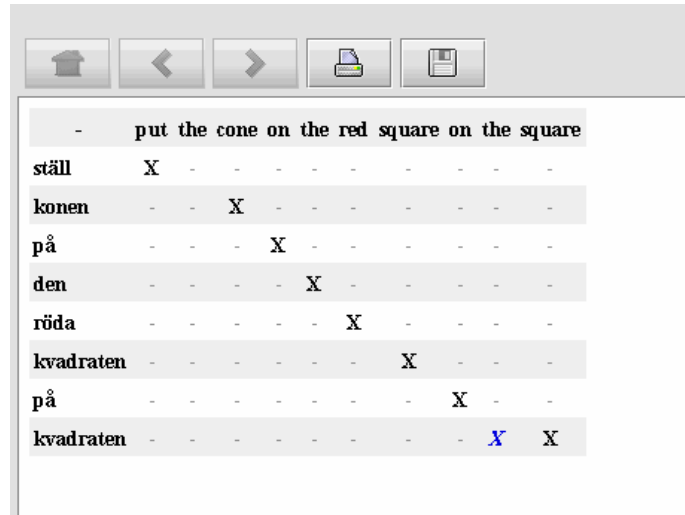


Figure 3.3 Alignment Area

3. **Function Area** includes all the buttons and parameters in I*TriX. All the functions and parameters can be set here. (Figure 3.4)

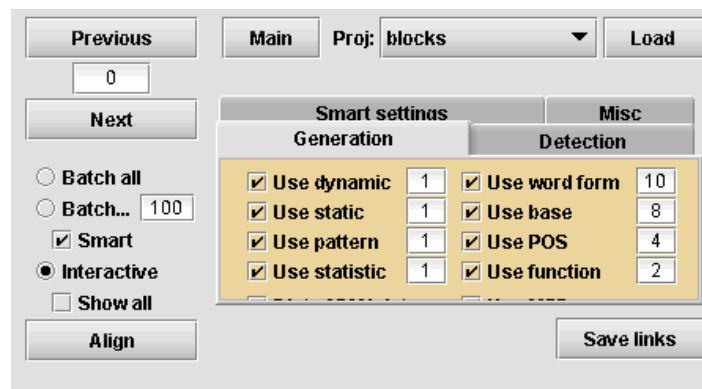


Figure 3.4 Function Area

In the function area, “**Main**” button selects the project directory. “**Proj**” area chooses different projects in Main directory. “**Load**” button starts to load the project into I*TriX. “**Previous**” and “**Next**” buttons are used for showing different sentence pairs of current project in the bitext segment area. After entering the sentence id in the “**sentence id area**”, which is the little number area between “**Previous**” and “**Next**”, the certain sentence pair will show up in the bitext segment area. Parameters can be set in the “**parameter area**”. There are several ways to set parameters. The way of choosing proper number in parameters to get good result will be discussed in detail later in section 5. There are three ways of aligning. “**Batch all**” means aligning all the sentence pairs in the current project at one time. “**Batch**” plus the number area following will let I*TriX align certain numbers of sentence pairs from sentence id in the “**sentence id area**”. “**Smart**” function means batching sentences from the shortest one to the longest one, and after the shorter sentences are aligned, alignment results will be added into a **resource** named “**auto-dynamic**”. This resource will be used in the longer sentences alignment to improve the file alignment result. “**Interactive**” just aligns one sentence, the current sentence in the “**sentence id area**”. The

alignment result will then be displayed in the “Alignment area”. If “**show all**” is selected, numbers will be shown in the “Alignment area”. Otherwise, only a cross will appear to show how the source and target sentence are linked, as in Figure 3.3. After finishing the alignment, the “**Save links**” button is used for saving the alignment results in link file format.

3.2.2 Input file formats in I*Trix

XML is the only available bitext format in I*Trix. Source and target texts at the sentence level encode in eXtensive Markup Language. This format includes information about Word Form, Base, POS and Function. Trying to load xml bitext as a project used in I*Trix, the supplied Document Type Definition: LIU-MONO.DTD must be put in the same folder with the xml file. This XML format is also the input format for I*Eval, which is the evaluation tool used in this project and which will be described in 3.3. And for “xml2txt” program, which will be mentioned later in Section 4.1.1, source file in XML will be converted into plaintext. The example of XML format is shown below.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE linCorpus SYSTEM "liu-mono.dtd">
<linCorpus>
  <linHeader></linHeader>
  <text>
    <body>
      <div id="d1">
        <p id="p1">
          <s id="s1">
            <w id="w1" relpos="1" base="show" pos="V" msd="IMP" func="main"
fa="&gt;0" stag="VA">Show</w>
            <w id="w2" relpos="2" base="all" pos="PRON" msd="" func=""
stag="NH">All</w>
          </s>
          .
          .
          .
        </p>
      </div>
    </body>
  </text>
</linCorpus>
```

3.2.3 Output file formats in I*Trix

The output file format in I*Trix is Link file format. In this report, this format appears several times. It is used as one of the input formats in I*Eval and the output format in Giza2Link program. For the gold standard used in this project, the format is also link file format. It is because the gold standard comes from another program I*Link, and the output of I*Link can be in link file format.

The format in link file looks like this:

```
0#(0|0|0|0|5)#(1|1|1|1|5)#(2|2|2|2|5)#(3|3|3|3|5)#(4|4|4|4|5)#(5|5|5|5|5)#
1#(0|1|0|0|5)#(-1|-1|16|16|5)#(2|2|2|2|5)#(3|3|1|1|5)#
2#(0|0|0|0|5)#(1|1|1|1|5)#(2|2|2|2|5)#(3|3|-1|-1|5)#(4|5|3|3|5)#
```

In each line of link file format, the first number is sentence id, which is integer and starts from 0. And then a “#” come. Each pair of bracket presents an alignment from source to target sentence. Alignments are separated by “#”. In the bracket, (|a|b|c|d|e|) implies in the source sentence, word id “a” up to word id “b” are linked to the words from word id “c” to word id “d” in the target sentence. The last number “e” can be only 4 or 5. 4 means the link comes from an auto alignment program, like I*TriX. 5 means the link comes from manually alignment program, like I*Link. Remarkable, either from “a” to “b” or from “c” to “d”, the words group are continuous. For example, if the fifth word in the source sentence wants to link to the fifth and seventh word together in the target sentence, it is impossible to express it in link file format.

3.3 Evaluation tool – I*Eval

I*Eval is developed as an evaluation tool for link results from I*Link and I*TriX. It is the only evaluation tool used in this project. I*Eval is developed at the Natural Language Processing Laboratory (NLPLAB), Department of Computer and Information Science at Linköping University, Sweden, with funding from The Swedish Research Council (Vetenskapsrådet) and The Swedish Agency for Innovation Systems (VINNOVA). I*Eval is implemented by Michael Petterstedt as a general tool for evaluating the link file format.

The input formats for I*Eval are XML file and link file. The output is evaluation report. The first interface of I*Eval shows in Figure 3.5.

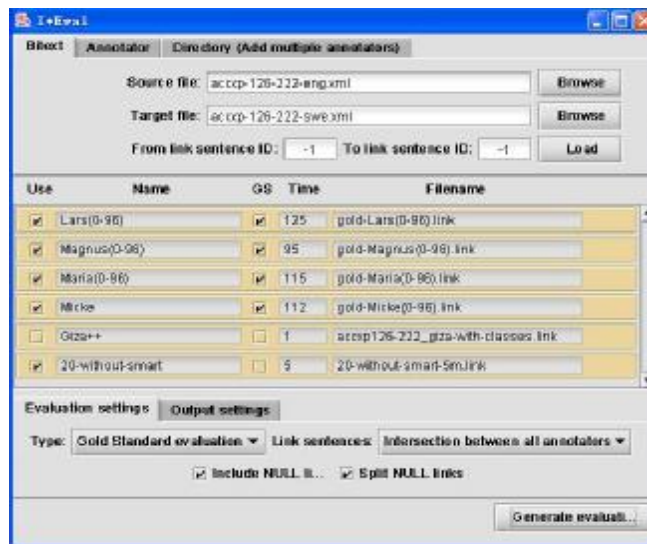


Figure 3.5 First interface of I*Eval

3. Main tools related in this project

In the “Bitext” tag, the source and the target files need to be chosen. The only available format here is XML. In the number area of “From link sentence ID” and “To link sentence ID”, if the default number –1 is kept, it implies using all the sentences in the source and target files. Otherwise, the specified numbers of sentences will be used to load only parts of sentences in the whole corpus. After pressing “Load” button, the second interface will show up. (Figure 3.6)

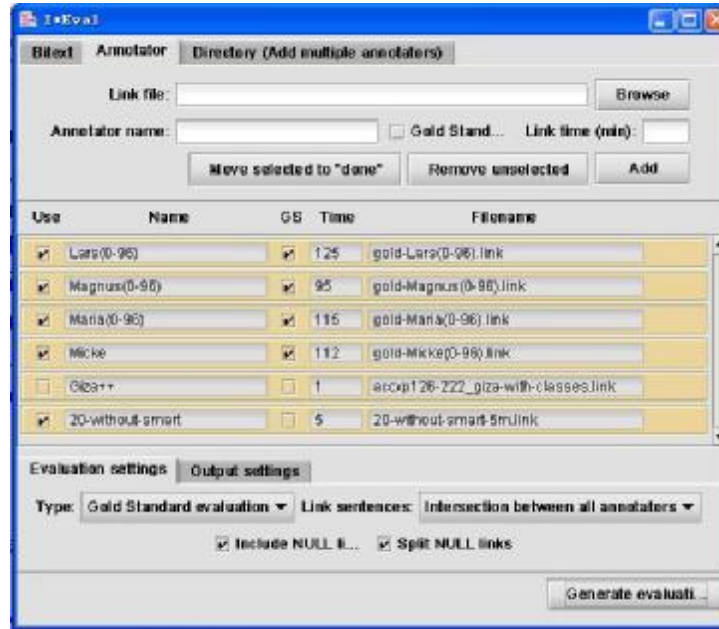


Figure 3.6 Second interface of I*Eval

The link files can be added here for evaluating. If the link file is a gold standard, the checkbox “Gold Standard” should be selected. The “Link time” area cannot be empty. It refers to the time to create the link file. After pressing the “Add” button, the current link file will be added into the link file part. At the same time, a “.linkmeta” file with the same filename as the current link file will be generated in the current directory. It contains the information which I*Eval need of this link file. The next time this link file needs to be loaded by I*Eval again, the information will be input automatically.

In the “Evaluation settings” tag, there are two checkboxes. One is “Include Null links”, if this item is selected, null links in the link file will be calculated as one token. Another one is “Split NULL links”. If this item is selected, all the group of words linked to null will be considered as several null links.

In the “Link sentences” part, two functions can be chosen, which are very useful when evaluating several link files. The “Intersection between all annotators” selects intersection of sentences IDs in all of the link files and compares the common sentences. If choose the other function “Only those in Gold standard(s)”, all the link files will obey the sentences IDs when they are compared in Gold standard.

Noticeable, two types of evaluation can be chosen. One type is “Similarity evaluation”. This is an evaluation comparing each pair of link files. It doesn't matter

Evaluation of Two Word Alignment Systems

if there is no gold standard in any of link files. An example report is shown in Table 3.1.

Start AB (NULL-LINKS INCLUDED)						
ID	Name	User links	Unique links	User type links	Unique type links	Calc. link time
A	Lars(0-96)	1560	150	843	101	123.5
B	Magnus(0-96)	1528	118	822	80	93.7
Summary						
Sentences						
	Base sentences:	96	Sents	(1-96)		
	Identical sentences:	53	Sents	(1, 3, 8, 11, 15-23, 26-28, 30, 32-34, 37-52, 54, 57, 66-67, 69, 71-72, 75, 80, 82-85, 87, 91, 93, 96)		
	Sentences similarity score:	55.2	%	(Identical sentences / Base sentences)		
Links						
	Means of links:	1544.0	Links	(Total links for all annotators / No of annotators)		
	Identical links:	1410	Links	(Union links among annotators)		
	Links similarity score:	91.3	%	(Identical links / Means of links)		
Type links						
	Means of type links:	832.5	Links	(Total type links for all annotators / No of annotators)		
	Identical type links:	742	Links	(Union type links among annotators)		
	Type links similarity score:	89.1	%	(Identical type links / Means of type links)		
	Means of type-token link ratio:	0.5		(Means of type links / Means of token links)		
Link & time statistics						
	Mean link time:	108.6	Min	(Total minutes / No of annotators)		
	Time per sentence:	1.1	Min	(Mean link time / Base sentences)		
	Links per sentence:	16.1	Links	(Means of links / Base sentences)		
	Links per minute:	14.2	Links	(Means of links / Mean link time)		

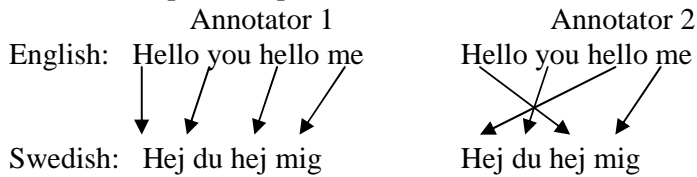
Table 3.1 Similarity Evaluation Report

Table 3.1 is a report compares with two link files.

“**User links**” means the number of word links in each link file. “**Unique links**” is the number of word links in one link file, which are different from in other link files.

“**User type links**” refers to the number of link types in every link file. Link types consider only on word form of links. “**Unique type links**” is the number of type links in one link file, which are different from in other link files.

Here is a simple example for these four definitions.



For Annotator 1, there are four “User links” and two “Unique links” because there are two links which are different from the links in annotator 2. “Hello” and “hello” are different but in one word form. So annotator 1 has 3 “User type links” and 0 “Unique type links” because there is no different link.

Start Giza++-without-classes (NULL-LINKS INCLUDED)

Partial enclosed by parenthesis

Full evaluation

A	1322	Sum(User links)
S	1002	Sum(True key links)
P	1702	Sum(Optional key links)
A i S	504.0 (557.52)	Sum(Intersect(User links, True key links))
A i P	587.0 (594.6)	Sum(Intersect(User links, Optional key links))
-		
Recall	0.5 (0.56)	A i S / S
Precision	0.44 (0.45)	A i P / A
AER	0.53 (0.5)	1 - ((A i S + A i P) / (A + S))
F	0.47 (0.5)	(2 * Precision * Recall) / (Precision + Recall)

MWU evaluation

A	35	Sum(User links)
S	109	Sum(True key links)
P	312	Sum(Optional key links)
A i S	0.0 (0.75)	Sum(Intersect(User links, True key links))
A i P	4.0 (3.29)	Sum(Intersect(User links, Optional key links))
-		
Recall	0.0 (0.01)	A i S / S
Precision	0.11 (0.09)	A i P / A
AER	0.97 (0.97)	1 - ((A i S + A i P) / (A + S))
F	0.0 (0.02)	(2 * Precision * Recall) / (Precision + Recall)

End Giza++-without-classes (NULL-LINKS INCLUDED)

Table 3.2 Gold standard evaluation report

“**Calc. Link time**” in the report calculates the time for the annotator to link common sentence links.

Evaluation of Two Word Alignment Systems

In the “**Summary**” part, “**Base sentences**” is the number of evaluation sentences. In the bracket behind that number, the sentences IDs are listed. “**Identical sentences**” is the number of sentences that have exactly the same links in the two link files.

Another type of evaluation is “Gold Standard evaluation”, which compares non-gold-standard link files with gold standard. If this type is chosen, then in all the link files, at least one must be gold standard. An example report in this type is shown in Table 3.2.

In Table3.2, “**True key links**” means the number of gold standard links. If there are several gold standard files, True key links are the intersection among them. “**Optional key links**” is all the gold standard links’ union. **Recall, Precision, AER** and **F-Measure** are four measurements that are used in this project to tell the evaluation results, which will be discussed in the next section. **MWU evaluation** means “Multiple word Units evaluation”. MWU evaluation is not the main evaluation in this project to be discussed.

4. The evaluation environment

4.1 Symmetrization

In order to evaluate the two systems fairly, symmetrization is very important. Because Giza++ and I*Trix systems are quite different on several parts, at least the same input data needs to be used during evaluation. With a certain corpora, input files, output files for these two systems need to be found out. In this part, how to make one corpus apply in different systems will be addressed.

4.1.1 Inputs symmetrization

No matter what format of bitext can be got originally, making the bitext be applied in two systems is a necessary step. As discussed before, the input in Giza++ is just plaintext. In I*Trix, XML format is the input file format.

If the original file is in XML format, using xml2txt program can convert the file to plaintext format. Xml2txt is a short java program implemented for this project. The purpose for this program is following the tag in the XML files, taking the <s> tag as a sentence in one line, and the <w> tag as a word. To separate each word, a Space will be used. After each sentence an Enter will follow to change to another line. Appendix C is the code for this program.

If the original file is in plaintext, several things need to be noted.

First, for some reason, each line starts with ##n## sometimes in plaintext. “n” is an integer number which presents the sentence id. This string will be treated as a word in both Giza++ and I*Trix, which will make a big influence for the system output. Trying to make things easier and clearer, it’s better to remove this string. “removenmb” can be used for this. It will automatically remove the sentences ID string in its input file and keep only sentences in its output file. This is a program which came from IDA/NLPLab.

Second, usually in plaintext the punctuation will follow the last word in the sentence without space, like this: “Today’s weather is good.” But if this sentence is put into Giza++, the system will considerate “good.” as a word, which is different from “good” and “.”. In I*Trix input file format XML, on the contrary, the punctuations always separates from the word, so all the full stops will be thought of as one word, not following with the last word in the sentence. For the sake of equitable evaluation, the punctuation problem needs to be considered. The easiest way for doing this is to use IFDG (Petterstedt 2003) to convert the plaintext into XML file. In this step, the punctuations are separated by a space from the word. This XML file can be used for I*Trix input. After that, use xml2txt program to convert this XML file back to plaintext. Because the xml2txt will follow the XML format, if a full stop has its own <w>, it will be thought of as a word. The punctuation problem will be solved after this

step. This kind of plaintext can be used as Giza++ input file because it is equal with the XML bitext.

4.1.2 Outputs symmetrization

The outputs from I*Trix are in link file format. There are several outputs from Giza++ as discussed before, but only “A3.final” is needed in this project. The evaluation tool, I*Eval, uses link file format as input. In order to use I*Eval, the only thing needed to do is to convert the “A3.final” into link file format is.

The java program “Giza2Link” is used here. This is a program implemented by Michael Petterstedt, who also implemented I*Trix. It is used for converting “A3.final” format into link file format. Noticeable, in “A3.final”, some sentences might be like this:

```
# Sentence pair (2) source length 26 target length 20 alignment score: 6.69883e-14
Kategorietiketter visas oftast över diagrammets x-axel , vilket dock kan variera
beroende-på vilken typ av diagram som du använder .
NULL ( { 17 } ) Category ( { 1 } ) labels ( { 2 } ) usually ( { 3 } ) appear ( { 4 } ) across
( { 5 } ) the ( { } ) x ( { 6 } ) axis ( { 13 14 } ) of ( { } ) the ( { } ) chart ( { } ) , ( { 7 } )
although ( { 8 } ) this ( { } ) can ( { 10 } ) vary ( { 9 11 } ) depending ( { 12 } ) on ( { } )
the ( { } ) type ( { 19 } ) of ( { 15 } ) chart ( { 16 } ) you ( { 18 } ) are ( { } ) using ( { } ) .
( { 20 } )
```

In the source language line, English word “vary” links to the ninth and eleventh words in the target language. The corresponding words in Swedish are “dock” and “variera”, which are words separated by another word “kan”. This kind of link is called **discontinuous link**. As discussed before, the link file format cannot handle discontinuous links. This is the toughest problem for the symmetrization of this two systems output.

Giza2link automatically removes the sentences including discontinuous links in A3.final file and adds the removed sentence ids into a list. If the link files from Giza++ system are compared with other link files only sentences left will be compared.

For example, for corpus accxp-126-222, if I*Eval compares the gold standard with one of I*Trix output, there are 1272 user links in I*Trix. But if I*Eval compares the gold standard with one of Giza++ output, only 1014 user links are found. This is because in the link file from Giza++, the whole lines of links that the sentences include discontinuous links are removed. In general, around 1/4 corpus of sentences will be removed because of discontinuous links.

4.1.3 Resources used

Some word alignment systems utilize extra resources, such as bilingual dictionaries and function word lists. The resources used by a particular system are valuable

information in the evaluation. To evaluate the two systems fairly, resources have to be stated clearly.

In I*TriX, there are several resources, such as dynamic, pattern, static and statistic. All of them can influence the I*TriX output result. On the other hand, in Giza++, the extra resource is optional, which is not included in this project. But Giza++ uses several Models to train the corpus. The output from the first model is used as the input for the second model, and so on. It is similar with using dynamic resources based on the training procession. Trying to symmetry two systems, all the settings about the resources in I*TriX are set like this:

- | Specified statistic resources are included;
- | Other resources are default files in the “common” folder in I*TriX;
- | There is no dynamic resource even in “common” folder. Just if choose “smart” function is chosen when running I*TriX, the auto-dynamic resource will be build up based on “smart”.

4.2 Participating Corpora

There are not a lot of resources in English-Swedish. Finding available corpora is always a problem especially a big corpora with proper gold standard. Considering different areas of content, different characteristics and different text size, this task turns to be more difficult.

Finally, three main corpora are used in this project. The characteristics for them are listed below:

- | Two of them are technology documents, and one is a normal document.
- | Two of them are small size corpora with around 100 sentences, the other one with more than 5000 sentences.
- | Two of them are with normal sentences, which mean the word frequencies are normal for every single word. The other one is a text with very few unique words. All the words have very high-appeared frequencies.
- | Two of them have punctuation, the other one does not have.

With these three corpora general results should be achieved. At the end of the project, some special corpora are created from these three main corpora to get some other results.

4.2.1 Blocks Corpus

Blocks is a corpus with 100 sentences. There are 683 words in the whole English text file, but unique words are only 40. In the Swedish text file, 64 unique words come from 607 words. There are a lot of repeat-words in that text. There is no punctuation.

4.2.2 Access XP 97 sentences Corpus

With this 97 sentences corpus the test steps are applied again to see if the conclusions are strong enough. *Access XP 97 sentences* corpus is a corpus with 97 sentences. It is a normal technical document with punctuations, which is not like *Blocks* with a lot of repeated words. There are 1798 words in the English text in which 499 words are unique. And 1515 words in the Swedish text and 535 of them are unique. The numbers of running words and the vocabularies are based on full-form words and the punctuation marks.

4.2.3 Access XP 5000 sentences Corpus

Access XP 5000 sentences is a corpus with 5382 sentences pairs. These are actually taken from a subset which contains pairs where the English sentences are between 2 and 20 words in length. There are 66051 words in the English text and 58495 words in the Swedish text. Among them, 4187 words are unique in the English text and 6304 words are unique in the Swedish text. The numbers of running words and the vocabularies are based on full-form words and the punctuation marks.

4.2.4 Corpora Summary

In Table 4.1, the data related to the three corpora are listed.

	Number of sent.	Words in Eng.	Unique Words in Eng.	Statistical ratio in Eng.	Words in Swe.	Unique Words in Swe.	Statistical ratio in Swe.
Blocks	100	683	40	17.1	607	64	9.5
Access XP 97	97	1798	499	3.6	1515	535	2.8
Access XP 5000	5382	66051	4187	15.8	58495	6304	9.3

Table 4.1

5. Analysis results

In this part, some evaluation tests with their results are presented.

5.1 Things that may influence the results

Before starting to discuss the tests and results, something during the whole process might influence the final results, which needs to be noted here.

5.1.1 In Giza++ system

Giza++ is a system with no graphical interface. Only command lines are used for running. In this project, all the input and output file formats in Giza++ are converted into the I*TriX format. During the converting process, some data from Giza++ may be lost and some of them may be changed. All of these may influence the results.

- I In Giza++ input file, plaintext format, only ASCII code can be handled. In the Swedish letters, ä, ö, å and their capital letters are non-ASCII codes. If they were kept in Giza++ inputs, in the output “A3.final”, every line which includes non-ASCII code will be cut by them, which means that the alignment information after non-ASCII code will be lost. To avoid this, before inputting the text file into Giza++, some strings have to be used to replace ä, ö, å and their capital letters. The strings are chosen arbitrarily, but it’s better to use long strings in case the chosen strings appear in the original text file. In this project, the following strings are used:

```

Ä---##A##
Ö---##O##
Å---00A00
ä---##a##
ö---##o##
å---00a00

```

After getting the “A3 final” file from the output, all the strings above should be changed back to Swedish letters. During two times of “Find and change” for these strings, some mistakes might be made which could influence the results.

- I Some sentences are truncated during running Giza++ because they are over the limitation of sentences length (61 words). Within the three corpora used in this project, *Blocks* only has short and simple sentences; *Access XP 5000* is a subset which contains pairs where the sentences are between 2 and 20 words in length. So only *Access XP 97* sentences might have this problem. In I*TriX, the sentences are complete. The different sentences source in the two systems will absolutely influence the result. This is a fact that cannot be changed. The only thing can be done is to keep this in mind when analysing the result.

- I From A3.final file converting to link file, links of some sentences are removed because of discontinuous links, as discussed in Section 4.1.2. This comes from different characteristics of the two systems. Nothing can be done to avoid this. When using I*Eval, selecting the “intersection between all annotators” function, then all the removed sentences won’t be compared. It’s not easy to tell if the removed sentences will change the result or not, but anyway, the corpora are changed because of this. So the influence might happen because of this.

5.1.2 In I*Trix system

The most important thing to influence I*Trix outputs is the setting of parameters. There are a lot of parameters that may have several combinations. Each way of setting up will lead to different results for different corpora. This will be discussed in detail with tests later in this section.

5.1.3 In I*Eval

Because there is an “intersection between all annotators” function in I*Eval, then how to choose the evaluate link files becomes a problem. Different ways of choosing will induce different show up reports.

The way of using I*Eval in this project is to choose only one evaluation link file and its correlative gold standard at the same time, which means only to compare the intersection sentences between gold standard and the current link file. If all relative link files are selected at one time, especially including Giza++ link file, links of many sentences will be removed and do not count for all the link files. For example, there are two link-files, T1, T2 come from I*Trix, two link files, G1, G2 come from Giza++, and one gold standard file. To see the evaluation report for T1, choose T1 and gold standard file. The next time, choose T2 and gold standard together. And so on. This way will keep the maximum link sentences in evaluation and still get reasonable results. Otherwise, if choose T1, T2, G1, G2 and gold standard together, only the sentence IDs both in G1 and G2 will be evaluated in T1 and T2.

5.2 Comparing the results from the two systems

From here, the evaluation tests are stated. The first two parts are discussion about parameter setting in I*Trix and Giza++ separately. From 5.2.3, different characteristics corpora are used for testing different evaluation purposes.

5.2.1 Parameter Setting in I*Trix

I*Trix is an automatic tool for creating and storing associations between segments in a bitext. Although it does automatic aligning, the very important advantage of this program is that the users can set a lot of parameters by themselves to choose the most suitable results. But just because of this point, I*Trix becomes a quite complicated program. Until now, there is no common rule for the users to tell generally which kind of parameter setting is the best for the specific corpus.

Before starting to set the parameters in I*Trix, there is something noticeable.

About Null links, in I*Trix, one word will be linked to Null link only if it has been linked to null links before in dynamic resources. For the moment, all the corpora for the evaluation project have no dynamic resources. The dynamic level In I*Trix is just influenced by “smart” function. So in all the tests following, the NULL links checkboxes are unselected.

Using “smart” function, it is almost sure that select the “smart” checkbox and “dynamic” together, I*Trix result will be better than not using “smart” function. So the “smart” and “dynamic” are selected in all the tests.

For all the corpora, static resources and pattern resources in “common” folder are applied. But each of them uses specific statistic resources.

Based on the align strategy, according to which Multiple Word Units are really important in the gold standard, for all the tests, MWU data are included. The number of MWU1 and MWU2 are set to 5 in order to get more MWU links in general.

Table 5.1 are the steps for testing I*Trix parameters. There are 20 test IDs, with 20 different ways of setting up. Because the Giza++ system is based on statistic principles, so the test ID in I*Trix start with only choose “statistic” and “wordform”. Note all the tests results that come from the Giza++ in this section are with word classes.

Test ID	dyn	Static	Pattern	Statistic	wform	BASE	POS	Func
1	1	0	0	1	1	0	0	0
2	1	0	0	1	1	1	0	0
3	1	0	0	1	1	1	1	0
4	1	0	0	1	1	1	1	1
5	1	0	1	1	1	0	0	0
6	1	1	0	1	1	0	0	0
7	1	1	1	1	1	0	0	0
8	1	1	1	1	1	1	1	0
9	1	1	0	1	1	1	1	0
10	1	1	1	1	10	1	1	1
11	1	1	1	1	1	10	1	1
12	1	1	1	1	1	1	10	1
13	1	1	1	1	1	1	1	10
14	10	1	1	1	1	0	0	0
15	1	10	1	1	1	0	0	0
16	1	1	10	1	1	0	0	0
17	1	1	1	10	1	0	0	0
18	1	1	1	1	10	8	4	2
19	1	1	1	1	1	1	1	1
20	1	1	1	1	10	8	4	0

Table 5.1

Two small corpora are tried first with this test table. The best parameters, which can be used in future I*Trix work to get better results, might be deduced from this test table.

1. With Blocks corpus

Blocks is a small corpus with 100 sentences. Specific Statistic resources for this corpus are used in the I*Trix. It applies a gold standard file with links of 100 sentences. During the process from the Giza++ output converting to link file format, there are 3 sentences that are removed because of discontinuous. So actually, with I*Eval “Intersection between all annotators” function, only links of 97 sentences are evaluated in each link file. The following table is the evaluation results (Table 5.2).

Test ID	Recall	Precision	AER	F-Measure	User links
1	0.78	0.86	0.19	0.82	547
2	0.79	0.86	0.17	0.82	558
3	0.79	0.86	0.17	0.82	558
4	0.79	0.86	0.17	0.82	558
5	0.81	0.86	0.17	0.83	568
6	0.78	0.86	0.18	0.82	549
7	0.81	0.86	0.17	0.83	570
8	0.95	0.97	0.04	0.96	587
9	0.85	0.87	0.14	0.86	589
10	0.89	0.91	0.1	0.9	592
11	0.88	0.9	0.11	0.89	592
12	0.71	0.82	0.24	0.76	517
13	0.48	0.63	0.45	0.54	460
14	0.81	0.86	0.17	0.83	570
15	0.81	0.86	0.17	0.83	570
16	0.81	0.86	0.17	0.83	570
17	0.81	0.86	0.17	0.83	570
18	0.94	0.96	0.05	0.95	592
19	0.79	0.86	0.17	0.82	558
20	0.93	0.95	0.06	0.94	589
Giza++	0.82	0.72	0.23	0.77	658

Table 5.2 Result of test steps of *Blocks*

The number in “User links” column cannot tell if the test result is better compared with other test results. As shown in Giza++ row, the “User links” is the largest, but the F-measure result is not the best one, and vice versa.

Here are the conclusions deduced from Table 5.2.

1. Test ID 1 and ID 2 only have difference on “BASE” level. F-measure has the same results in these two tests. Just Recall and AER in ID 2 are better than in ID 1. It shows that selecting “BASE” level might improve I*Trix result for this corpus, but the influence is very small.

2. Test ID 2, 3 and 4 have exactly the same results. It means that at least on statistical resource of this corpus, selecting POS and Function does not make progress on the results.

3. From ID 5, only “wordform” is selected among all the levels. The purpose for test ID 5, 6 and 7 is to find out the clue about how dynamic, static, pattern and statistic resources influence the I*Trix results. Comparing with ID 5 to ID 6, they are different only on “static” and “pattern”. The result of ID 5 is better than the result of ID 6, and both of the results are better than ID 1. ID 7 has almost the same result as ID 6. As can be expected, both static and pattern resources can make I*Trix result well in this corpus at wordform level. Static resources give more influence.

4. In Table 5.2, ID 8 has the best result, which is not surprising. It matches the guess that the function can only give little improvement or sometimes even worse influence. It also shows that at least one of static and pattern resources can improve the results with its POS level. Compared with ID 9, ID 3 and ID 8 we can conclude that both pattern and static can raise the result at POS level in this corpus.

5. From ID 10, a big number 10 is tried instead of all “1” in the parameters. ID 19 is a base test in which all the parameters are 1. In ID 10, with “10” in Wordform level, comparing with the base test ID 19, the result increases $(0.9-0.82)/0.82=9.76\%$. With the “BASE” level as 10 in Test ID 11, the result makes $(0.89-0.82)/0.82=8.54\%$ progress. ID 12 gets $(0.76-0.82)/0.82=-7.32\%$, which means that the big number in “POS” makes the result worse. Same as ID 13 in the Function level, which has the worst result in the whole table, it has decreased the result $(0.54-0.82)/0.82=-34.15\%$.

6. ID 14, 15, 16 and 17 have exactly the same results again, which means that the big number 10 in different resources does not have a big influence at Wordform level in this corpus. Because Wordform is the most basic level, set all the numbers in resources as 1 will make I*Trix easy.

7. ID 18 is an ideal parameter setting deduced from the test results above. Based on different percentage of influence, the numbers in wordform highest as 10, and “BASE” 8, for POS and function number 4 and 2 are given separately. Although the result here 0.95 is not the best in the whole table, it is also a peak point within all the parameter setting tests.

8. Another interesting result from test ID 20 is also shown in Table 5.2. From all the conclusions mentioned above, it’s very easy to guess that to choose “function” can only make the result worse. So in test ID 20, “Function” is set as 0. The result from ID 20 proves that it is a good choice also. Although the results in this test are also good, almost the same as in ID 8 and 18, still, it is just a peak point but the best result.

9. Most of the F-measure results from I*Trix, except ID 13, are better than the results from Giza++. The best result from I*Trix is $(0.96-0.77)/0.77=24.68\%$ better than from Giza++, which is quite remarkable.

Above, later when using *Blocks* corpus, test ID 8 is the choice of I*Trix parameter setting. And for this corpus, I*Trix handle it better than Giza++.

2. With Access XP 97 sentences corpus

With this 97 sentences corpus, the steps (in Table 5.1) are applied again to see if the conclusions got with *Blocks* corpus are strong enough. Specific Statistic resources are applied for this corpus in I*Trix. There are four different gold standard files and each of them has links of 97 sentences. This means in I*Eval, the True key links are the intersection of these four gold standard files. There are links of 30 sentences are removed in the Giza++ output file. Table 5.3 lists the evaluation results.

Test ID	Recall	Precision	AER	F-Measure	User links
1	0.29	0.42	0.65	0.34	953
2	0.31	0.44	0.63	0.36	993
3	0.31	0.44	0.63	0.36	993
4	0.31	0.44	0.63	0.36	993
5	0.32	0.45	0.62	0.37	1031
6	0.32	0.46	0.62	0.38	1001
7	0.35	0.47	0.6	0.4	1068
8	0.4	0.44	0.58	0.42	1313
9	0.43	0.47	0.55	0.45	1296
10	0.43	0.45	0.56	0.44	1350
11	0.42	0.45	0.56	0.43	1344
12	0.38	0.4	0.61	0.39	1344
13	0.25	0.29	0.73	0.27	1249
14	0.35	0.47	0.6	0.4	1068
15	0.36	0.47	0.59	0.41	1070
16	0.36	0.48	0.59	0.41	1067
17	0.32	0.43	0.63	0.37	1063
18	0.41	0.44	0.57	0.42	1343
19	0.4	0.43	0.59	0.41	1335
*20	0.41	0.46	0.56	0.43	1272
Giza++	0.47	0.41	0.56	0.44	1014

Table 5.3

From the first glance on Table 5.3, the results are basically similar with the conclusions from Table 5.2: 8, 18 and 20 are good, and ID 13 has the worst result. Still, there are some differences.

1. According to the result of Test ID 1, 2, 3 and 4 in Table 5.3, the first 2 conclusions drawn from Table 5.2 can be retrieved as well. The “BASE” level makes positive effect of I*Trix result, and “POS” and “Function” have no influence in this corpus.
2. The results in Test ID 5,6,7 plus ID 1 clearly show that selecting Static, Pattern and Statistic together can make better effect than just selecting one or two of them in this corpus.
3. Different from the result in Table 5.2, ID 9 is the best result in this table. It shows that, in most cases choosing “Function” doesn’t improve the result, and sometimes, even Pattern might make the result worse as showed in this corpus.

4. In comparison with the basic test ID 19, test ID 10 makes the result increase $(0.44-0.41)/0.41=7.32\%$. ID 11 makes $(0.43-0.41)/0.41=4.88\%$ progress. ID 12 has $(0.39-0.41)/0.41=-4.88\%$ worse, and ID 13 is the worst result in the whole table, which is $(0.27-0.41)/0.41=-34.15\%$ worse than ID19. Because the percentage numbers are close to the numbers from Table 5.2, the parameters used in Test ID 18 might also be good to use in this corpus.

5. Test ID 14, 15, 16 and 17 are not exactly the same in this corpus. ID 17 is the worst one among these four tests. It shows that bigger number in “Statistic” even makes a negative effect. The characteristics of the corpus might be one reason. Note that in *Blocks* corpus, the statistical ratios are very high. *Access XP 97 sentences* corpus has just normal numbers of repeated words. It is understandable that “Statistic” makes greater positive effect in *Blocks* than in *Access XP 97 sentences*. For the results in these four tests are close to each other, setting Dynamic, Static, Pattern and Statistic as “1” can be admitted to achieve good results in this corpus.

6. Test ID 18 give a peak point result. From one point of view, this is a generally satisfying setting of parameters. Noticeable, ID 20 in this corpus has better result than ID 18, which is different from in Table 5.2. This result is perhaps because of different characteristics of corpus. ID 18 and ID 20 are parameters settings needed to be further explored.

7. The result from Giza++ in this corpus is almost the same as from I*Trix. F-measure from Giza++ is 0.44 while the maximum F-measure from I*Trix is 0.45. They can be considered the same.

Although Test ID 9 has the best number of result in this table, considering most of the time, Pattern makes improvement for the result, and ID 18 and 20 are always give remarkable good results for both of the two corpora, a parameter setting result will be chosen from ID 18 and 20. In this table, test ID 20 has bigger number of result compared with ID 18, so in later corpus using in I*Trix, ID 20 is the chosen one for this corpus.

3. With Access XP 5000 sentences corpus

Finally, we can try to apply the parameters on a big sentences corpus, *Access XP 5000 sentences*. Specific Statistic resources are applied in I*Trix. For the link file from the Giza++ output, 1095 sentences are removed because of discontinuous links. One manual link file with links of 594 sentences is used as gold standard. Not all the sentences ids are continuous within this gold standard.

For the reason that running the I*Trix with big corpus of around 5000 sentences pair always takes quite a long time, there are only two test IDs applied here. With the test ID 18 and 20, which are two generally good settings of parameters, the results are shown in Table 5.4.

Test ID 20 has a little better result than ID 18. For the normal technical article like *Access XP help document*, one conclusion might be drawn that select “Function” is better than unselect it. So in the following test, all the I*Trix results for Access

corpora are tested with the parameter ID 20. All the tests on *Blocks* corpus use Test ID 8.

Test ID	Recall	Precision	AER	F-Measure	User links
18	0.64	0.7	0.33	0.67	5420
*20	0.65	0.72	0.32	0.68	5333
Giza++	0.75	0.59	0.34	0.66	5714

Table 5.4

5.2.2 Word classes in Giza++

In the process of running Giza++, the step of running “mkcls” is optional. “mkcls” is a separate package of Giza++. It is a tool to train word classes by using a maximum-likelihood-criterion. The resulting word classes are especially suited for language models or statistical translation models. In this part, how word classes influence the alignment result is addressed.

When Giza++ starts to run, for Hmm training and Viterbi training, vcb.classes files are looked for. If the system finds that those classes files come from “mkcls”, it will use the result for further training. Otherwise, Giza++ can also run without those classes files. It is interesting to know how different the results might be with and without classes files. So the tests are made as following.

1. The corpora to apply without-classes-test are *Blocks* and *Accxp 97* sentences. The results shown in Table 5.5 compare Giza++ result with and without classes.

Corpus	Word Classes	Recall	Precision	AER	F-Measure	User links
Blocks	Yes	0.82	0.72	0.23	0.77	658
	No	0.84	0.75	0.21	0.79	673
Accxp 97	Yes	0.47	0.41	0.56	0.44	1014
	No	0.5	0.44	0.53	0.47	1322

Table 5.5

2. For corpora contains different sentences numbers, *Access XP 500, 1000, 2000, 4000, 5000*, the tests with and without classes files are also applied. With this table, in order to see the evaluation results with different corpora’s size clearly, only F-Measure numbers are shown.

Word Classes	Size of training corpus				
	500	1000	2000	4000	5382
Yes	0.59	0.67	0.65	0.67	0.66
No	0.58	0.65	0.64	0.65	0.66

Table 5.6

From all the tests in Table 5.5 and 5.6, a conclusion that there is no obvious improvement appeared might be drawn as a result of word classes. A possible reason for this lack of improvement is that the word classes themselves cannot be estimated

reliably using a small training corpus: in Table 5.6, even if the corpora sizes are increasing, the results are still the same as with small size corpus.

The same conclusion also comes from Och & Ney (2003). In 6.6 (P41) of this article, “alignment models depending on word classes” is addressed. From the test table, they observed “no significant improvement in performance as a result of including dependence on word classes when a small training corpus is used”. “When a large training corpus is used, however, there is a clear improvement in performance on both the Verbmobil and the Hansards tasks.” The “large training corpus” mentioned here refers to 34K sentences pairs in Verbmobil task and 1470K sentences pairs in Hansards tasks. Even the largest corpus size with 5382 sentences pairs is very small compared with them.

Although there is no obvious improvement when using word classes, and in some tests using word classes even leads to worse results, word classes is still used in this project when running Giza++ to represent that this is a complete system.

5.2.3 Different sizes of corpora

From this section, different kinds of corpora are discussed. In 5.2.3 more attentions are paid to reveal if the difference of the corpora’s sizes will affect the results. Later, in 5.2.4 a repeated corpus will be generated, and in 5.2.5 tests will be applied on monolingual corpora.

From Table 5.3 we can see that, the F-measure number is around 0.4. But in Table 5.4, the F-measure number is around 0.7. *Access XP 97* sentences and *Access XP 5000* sentences come from the same document, they only have different sizes of the corpus. Why the F-measure results have a quite big difference? Is this because of the corpora size, or is this because of some other reasons? These are interesting things to find out through tests.

In this project, different sizes of Access XP corpora are tested. In Table 5.7 the corpora and some characteristics data are listed.

Corpus	No. of Sent.	No. of GS sent.	English			Swedish		
			No. of Words	No. of Unique	Stat. ratio	No. of Words	No. of Unique	Stat. ratio
100	100	11	1082	333	3.2	925	359	2.6
500	500	55	5965	1152	5.2	5168	1378	3.7
1000	1000	110	12240	1624	7.5	10760	2051	5.2
2000	2000	221	23994	2264	10.6	20790	3061	6.8
4000	4000	441	49124	3650	13.5	43506	5305	8.2
5000	5382	594	66051	4187	15.8	58495	6304	9.3

Table 5.7

In Table 5.7, the corpora including 100, 500, 1000, 2000, 4000 and 5382 sentences are enumerated separately. This table shows that with the increase of the corpus size, for the same article, although the No. of words and No. of Unique words are

Evaluation of Two Word Alignment Systems

increasing, the statistical ratios are also increasing in both English and Swedish texts. The main reason is because in the same article, some topics are discussed around some high frequency words. With the increasing of the corpus size, some certain words are repeated again and again. In addition, the number of words in Swedish are always less than in English, but the numbers of unique words are higher than in English. So the statistical ratios in Swedish text are always lower than in English text.

All the corpora are subsets of the start of the corpus *Access XP 5000 sentences* corpus. The maximum number of gold standard sentences is 594, and the maximum number of sentences is 5382. In order to make different size of corpus comparable, different numbers of gold standard sentences are used. In different corpora, the percentages of gold standard sentences are the same. Two ways of picking up gold standard from 594 sentences link are used. One way is to choose gold standard sentences from the beginning of 594 sentences, while the other way is to choose them from the end of the gold standard if only they can be used. For example, there are links of 110 gold standard sentences in 1000 sentences corpus, which are gold standard sentences selected from No. 484 to No 594. But for 100 sentences corpus, the proper gold standard sentences are from No. 89 to 100.

Table 5.8 shows the evaluation results with the gold standard selected from the beginning of 594 sentences (all available gold standard of this corpus), and Table 5.9 shows from the end. All the results applies the parameter setting test ID 20 in I*Trix.

Test ID	Test Corpus	Recall	Precision	AER	F-Measure	User links
I*Trix	100	0.53	0.54	0.47	0.53	104
	500	0.66	0.71	0.32	0.68	471
	1000	0.65	0.73	0.31	0.69	881
	2000	0.63	0.68	0.34	0.65	1908
	4000	0.64	0.71	0.32	0.67	3810
	5000	0.65	0.72	0.32	0.68	5333
Giza++	100	0.63	0.48	0.46	0.54	88
	500	0.68	0.52	0.41	0.59	475
	1000	0.75	0.61	0.33	0.67	926
	2000	0.73	0.58	0.35	0.65	1907
	4000	0.76	0.6	0.33	0.67	3897
	5000	0.76	0.6	0.33	0.67	5717

Table 5.8 With GS from the beginning

In Table 5.8, random results are achieved from both I*Trix and Giza++. Both of them have the best results with corpus 1000 sentences and the worst results with corpus 100 sentences. It is not wise to draw the conclusion that the results have no relationship with the corpus size. This kind of results can only show that the 1000 sentences corpus with its gold standard has a better fit compared with the other size of corpus.

From Table 5.9 the guess above can be proven again, that the size of the corpora is not the only thing that can decide the result of word alignment system. In this table, with the test corpus of 100 sentences, the best results are achieved while with the same size in Table 5.8 the worst results are got. It only shows that in this way of choosing gold standard, the 100 sentences corpus has a better fit than other size.

Test ID	Test Corpus	Recall	Precision	AER	F-Measure	User links
Parameter ID 20 from I*Trix	100	0.68	0.81	0.26	0.74	94
	500	0.64	0.69	0.33	0.66	568
	1000	0.64	0.74	0.31	0.69	1100
	2000	0.66	0.73	0.31	0.69	2082
	4000	0.65	0.73	0.31	0.69	4032
	5000	0.65	0.72	0.32	0.68	5333
Giza++	100	0.83	0.75	0.21	0.79	64
	500	0.68	0.55	0.39	0.61	544
	1000	0.74	0.6	0.34	0.66	1009
	2000	0.73	0.58	0.35	0.65	2181
	4000	0.75	0.6	0.33	0.67	4272
	5000	0.76	0.6	0.33	0.67	5717

Table 5.9 With GS from the end

In Och & Ney (2003) “A Systematic Comparison of Various Statistical Alignment Models”, the size increasing made very significant improvement of the results. One guess for the difference in the two articles is that in Och & Ney (2003), they used very big corpora. The biggest size, 5832 sentences, in this project is just a small number in that article. So perhaps the influence of corpus size can only be noticed with very big corpora.

5.2.4 Repeated corpora

From the discussion above, the characteristics of the corpus sometimes influence the result a lot. Statistical resources are the only kind of specific resources used in this project. How the statistical characteristic of a corpus influences the evaluation result is therefore worth to be observed.

A test corpus is created for this purpose. The name of the corpus is “repeated-blocks-1000”. It is a corpus with 1000 sentences pairs. The content of the text is a 10-times-repeat of *Blocks* corpus. *Blocks* corpus has 683 words and 39 unique words in the whole English text file. In this repeated corpus, there are 6830 words, but still 39 unique words. In the Swedish text file of this repeated corpus, 63 unique words come from 6070 words. This kind of corpus might spread all the influence of statistical resource because it has very high Statistical ratio. See in Table 5.10.

Corpora	No. of Sent.	English			Swedish		
		No. of Words	No. of Unique	Statistical ratio	No. of Words	No. of Unique	Statistical ratio
Blocks	100	683	39	17.5	607	63	9.6
Repeated-Blocks-1000	1000	6830	39	175	6070	63	96

Table 5.10

For this repeated corpus, a specified complete gold standard is applied, which is a gold standard including links of 1000 sentences. The evaluation results from Giza++ (with classes) and I*Trix tests shown in Table 5.11.

Test ID	Recall	Precision	AER	F-Measure	User links
Giza++	0.73	0.59	0.35	0.65	7108
1	0.81	0.86	0.17	0.83	5768
2	0.81	0.86	0.17	0.83	5768
3	0.81	0.86	0.17	0.83	5768
4	0.81	0.86	0.17	0.83	5768
5	0.81	0.86	0.17	0.83	5768
6	0.84	0.86	0.15	0.85	5948
7	0.84	0.86	0.15	0.85	5948
*8	0.94	0.96	0.05	0.95	6008
9	0.85	0.87	0.14	0.86	6028
10	0.88	0.9	0.11	0.89	6028
11	0.88	0.89	0.12	0.88	6028
12	0.71	0.83	0.23	0.77	5278
13	0.49	0.64	0.45	0.56	4709
14	0.84	0.86	0.15	0.85	5948
15	0.84	0.86	0.15	0.85	5948
16	0.84	0.86	0.15	0.85	5948
17	0.84	0.86	0.15	0.85	5948
*18	0.94	0.96	0.05	0.95	6028
19	0.79	0.88	0.17	0.83	5518
20	0.93	0.94	0.06	0.93	6028

Table 5.11

From Table 5.11 some conclusions might be drawn.

- I Almost all I*Trix results are better than Giza++ with classes results except the worst one in ID 13. It is the same as the results in *Blocks* corpus. For the value of F-measure in this repeated corpus, most of them have around 0.2 higher results. It might indicate that at least for high statistical ratio corpus, it's easier to get better results with I*Trix.
- I Compared with Table 5.2, in Table 5.11, not only ID 2, 3 and 4, but also ID 1,2,3,4 and 5 all have the same results. It might reveal that in high statistical ratio corpus even the "BASE" level in Statistic resources has a very small influence on the results.
- I The results in ID 14, 15, 16 and 17 are exactly the same. In *Blocks* corpus, these four tests also have the same results. Different from the guess, the specific statistical resources do not improve the results for extreme high statistical ratio corpus.
- I The best result appears in ID 8 and ID 18. The highest F-measure number in this

table is 0.95, which is almost the same as the highest number in Table 5.2. All the other result numbers are also nearly the same as in *Blocks* corpus.

- l The discussion shows that with the same number of unique words, no matter how big the corpus is, the evaluation results are rarely different by using I*Trix.
- l Applying this repeated corpus in Giza++, it gets worse results compared with *Blocks* corpus. A conclusion might be drawn that Giza++ is less suitable to handle high statistical ratio corpora.

5.2.5 Monolingual corpora

Until now, all the corpora used are bilingual, which means that they contain English and Swedish language in source and target files. Trying to see if these two alignment systems are reliable, stable, and what is the difference when they align, the simplest test is to use monolingual with the same content in source and target text files.

A monolingual corpus is created for this reason as shown in Table 5.12. In this monolingual corpus, the source document is in English while the target document is exactly same as the source file.

Corpus Name	No. of sentences	English Source (Same as in Target)		
		No. of Words	No. of Unique	Statistical ratio
Blocks-eng-eng	100	683	39	17.5

Table 5.12

The evaluation results come from I*Trix and Giza++ shown in Table 5.13.

For Giza++ and most of the I*Trix test results, the F-measure is 1, which means the alignment files are exactly same as the gold standard. Gold standard for this corpus was linked manually from one word in source document to the same single word in target document. So the results show that both of the system can handle this simplest situation exactly what the annotators expected. From one point of view, it shows these two systems are reliable and stable basically.

Noticeable, not all of the tests in I*Trix reach the ideal result. It proves again the importance for studying parameter setting before using I*Trix. And if look up in the link files, sometimes the alignment will be linked like this way in I*Trix:

```
1#(|0|1|0|1|4|)#(|2|3|2|3|4|)#(|4|5|4|5|4|)#
```

Evaluation of Two Word Alignment Systems

For some reasons, it does not link 0->0, 1->1..., instead, it links with multiple words 0,1->0,1. Actually, these results cannot be said wrong. It just differs from the gold standard.

This test also emphasizes the essentials of the gold standard. Gold standard is always quite tricky in word alignment systems. It's difficult to tell if the alignment result is good or not because based on different link strategies and different annotators, gold standard alignments can be always different.

Used system	Recall	Precision	AER	F-measure	User links
Giza++	1	1	0	1	671
1	1	1	0	1	671
2	1	1	0	1	671
3	1	1	0	1	671
4	1	1	0	1	671
5	1	1	0	1	671
6	1	1	0	1	671
7	1	1	0	1	671
8	0.42	0.59	0.51	0.49	477
9	1	1	0	1	671
10	0.99	1	0	0.99	669
11	0.99	1	0	0.99	669
12	0.16	0.28	0.79	0.2	390
13	0.2	0.33	0.76	0.25	401
14	1	1	0	1	671
15	1	1	0	1	671
16	1	1	0	1	671
17	1	1	0	1	671
18	0.9	0.95	0.08	0.92	637
19	0.28	0.44	0.66	0.34	430
20	1	1	0	1	670

Table 5.13

5.3 Speed of two systems

The running environment for this project is UltraSPARC-IIi 440MHz CPU, 256MB (50ns) memory and the system is SunOS Release 5.8 Version Generic-108528-15 64-bit. For all the tests above, the speed of I*Trix is much slower than for Giza++ system. In general, using Giza++, the average speed is 0.1~0.2 seconds per sentence. For I*Trix, the average speed is 1~2 seconds per sentence. For example, for *Access XP* 5382 sentences, using Giza++ the speed is around 10 minutes, but with I*Trix, the speed is around 150 minutes. Giza++ speed is almost 10 times of the speed of I*Trix.

6. Summary and Conclusions

6.1 Summary

Two word alignment systems are introduced in this paper.

The input format in Giza++ is plaintext, while the input format in I*Trix is bitext in XML format. Trying to symmetrize these two systems, a program “xml2text” is developed to convert XML format into plaintext. To be able to convert from plaintext into XML format IFDG is used.

Giza++ uses command line. “Word classes” is an optional step that can be skipped. I*Trix has a beautiful interface, but the parameters in it are quite complicated and need to be studied carefully before using.

The output file in Giza++ used in this project is “A3 final”. The output file in I*Trix is link file. In order to use I*Eval as the evaluation tool, which also uses link file as its input, converting A3.final file into link file format is a necessary step. The program “Giza2link” is used here.

Three corpora are applied for the evaluation of the two systems. A detailed study about I*Trix parameters is made. In particular, two corpora tried the Giza++ results with and without word classes. For different corpus sizes, for low statistical ratio corpus and for monolingual corpus the evaluations are made.

6.2 Evaluation results related to parameter setting

From the I*Trix parameter setting study some conclusions can be made. For normal corpus both test ID 18 and ID 20 can achieve good results. For high statistical ratio corpus, ID 18 has better results than ID 20. For lower statistical ratio corpus, on the way round, ID 20 usually can get better results. From the study of the Giza++ parameter setting, which is “word classes”, the conclusion can be deduced that word classes cannot influence the results a lot, especially for small corpus.

6.3 Evaluation results related to corpora

For Giza++, it's easier to deal with the corpus that does not lie on specific resources. *Blocks* use specific statistical resources, so Giza++ gets worse results than I*Trix.

Using Giza++ is better for big corpora because the speed is much faster. For example, for corpus *Access XP 5000*, the result from Giza++ is almost the same as the result from I*Trix, but Giza++ is 10 times faster. Big corpora with word classes can lead to better results.

For small and high statistical ratio corpus or corpus with specific resource, I*Trix is a better choice.

Depending on different gold standard link strategies, both Giza++ and I*Trix might be a good choice for the same corpus.

6.4 Strengths and weaknesses

The two word alignment systems, Giza++ and I*Trix, have the same purpose. But they are all different in the way of running, the interface, and the file formats they can handle and so on. It's impossible and unfair to say which system is better. What can be summarized is which system has more strengths or weaknesses in certain situations.

Giza++ is a complete word alignment system. It follows the statistical machine model to deal with the data. The running speed for Giza++ is very fast, and the more training, the better results can be achieved. Although the running is from command line, it is still easy to learn. The weakness for Giza++ is that only "word classes" is an optional step. The parameters that can be changed by the users are very few. And because Giza++ implemented by using C, it can only use in Unix.

I*Trix, on the other hand, is not a complete software for independent using. The purpose of developing it is just a middle step in other programs. Although it has a good interface, still, setting the parameters and understanding all the functions are quite complicated. But I*Trix can deal with different corpus for different parameters very carefully. In particular, I*Trix can include specific resources for every corpus which might improve the result.

Giza++

- + Fast speed
- + Better for big corpora
- + Easy to start running
- No interface
- Only run on Unix
- Few setting ways

I*Trix

- + Better for high statistical ratio corpora
- + Better for corpora with specific resources
- + Different setting for different corpora
- + Run on several OS
- + Friendly interface
- Difficult to start running
- Speed problem

6.5 Future work

The results from this project might be useful for IDA/NLPLab for future research.

If complete gold standard for bigger corpus can be found or be made in the future, some evaluations on big corpus can be tested to achieve more precise and reliable results. In this project, the biggest corpus, with 5382 sentences pairs, is still very small for some tests.

In this project, only statistical resources are consisted for each corpus as specific resource. For the future work, other specific resources might also be introduced, such as dynamic resources.

This project only tests on English-Swedish corpora. In the future some other language might be included.

How these two systems deal with multiple words alignment is also interesting to know. This has not been paid enough attention to in this paper.

Evaluation of Two Word Alignment Systems

7. References

- Ahrenberg, L., M. Merkel, and M. Andersson (1998). A simple hybrid aligner for generating lexical correspondences in parallel texts. In Christian Boitet and Pete Whitelock, editors, Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL) and the 17th International Conference on Computational Linguistics (ACL/COLING), pages 29-35, Montreal, Canada, 1998. Morgan Kaufmann Publishers.
- Ahrenberg, L., M. Merkel, A. Sångvall Hein, and J. Tiedemann (1999). Evaluation of LWA and UWA. Technical Report 15, Department of Linguistics, Uppsala University, Uppsala, Sweden.
- Ahrenberg, L., M. Andersson, M. Merkel and M. Petterstedt (2000). I*Link - a graphical user interface tool for creating and storing associations between segments in a bitext. Natural Language Processing Laboratory (NLPLAB), Department of Computer and Information Science, Linköping University, Sweden.
- Baker, M. (1995). Corpora in Translation Studies – An Overview and Some Suggestions for Future Research. *Target* 7 ((2)): 223-243.
- Brown, P. et al. (1990). A statistical approach to machine translation. *Computational Linguistics* 16(2), 1990:79-85.
- Brown, P.F., J.C. Lai, and R.L. Mercer (1991). Aligning sentences in parallel corpora. Proceedings from the 29th Annual Meeting of the Association for Computational Linguistics (ACL-91): 169-176.
- Brown, P., S. Della Pietra, V. Della Pietra, and R. Mercer (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2).
- Francis, W. M. and H. Kucera (1964). *Brown Corpus Manual of Information*. Department of Linguistics, Brown University. Also available at <http://khnt.hit.uib.no/icame/manuals/brown/>.
- Gale, W. and K.W.Church (1991). A program for aligning sentences in bilingual corpora. Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics: 177-184.
- Hiemstra, D. (1998). Multilingual domain modeling in Twenty-One: Automatic creation of a bi-directional translation lexicon from a parallel corpus. In Peter-Arno Coppens, Hans van Halteren, and Lianne Teunissen, editors, Proceedings of the 8th Meeting of Computational Linguistics in the Netherlands (CLIN), number 25 in *Language and Computers: Studies in Practical Linguistics*, pages 41-58, Nijmegen, The Netherlands. Rodopi, Amsterdam, Atlanta.

Evaluation of Two Word Alignment Systems

- Isabelle, P. (1992). Bi-textual aids for translators. In Proceedings of the 8th Annual Conference of the UW Centre for the New OED and Text Research, pages 76-89, University of Waterloo, Waterloo, Canada.
- Kevin K. et al. (1999). The EGYPT Statistical Machine Translation Toolkit. <http://xbean.cs.ccu.edu.tw/~dan/oodbResearch/ConceptLearning/OchStochGrammarLearner/The%20EGYPT%20Toolkit%20Distribution%20Web%20Page.htm>
Center for Language and Speech Processing at Johns-Hopkins University (CLSP/JHU).
- Melamed, I.D. (1995). Automatic Evaluation and Uniform Filter Cascades for Inducing N-Best Translation Lexicons. Proceedings of the Third Workshop on Very Large Corpora. Cambridge: 184-198.
- Melamed, I.D. (1998). Annotation style guide for the Blinker project, version 1.0. IRCS Technical Report 98-06, University of Pennsylvania, Philadelphia, PA, 1998.
- Merkel, M. and L. Ahrenberg (1999). Evaluating Word Alignment Systems. Linköping University, Linköping, Sweden.
- Merkel, M. (1999). Understanding and enhancing translation by parallel text processing. Linköping Studies in Science and Technology. Dissertation No. 607. Linköping University. Dept. of Computer and Information Science.
- Merkel, M., L. Ahrenberg and M. Petterstedt (2003). Interactive Word Alignment for Corpus Linguistics. Proceedings of Corpus Linguistics 2003. UCREL Technical Paper No 16.
- Mihalcea, R. and T. Pedersen (2003). An Evaluation Exercise for Word Alignment. In Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond.
- Och, F.J. and H. Ney (2000a). A comparison of alignment models for statistical machine translation. In Proceedings of the 18th International Conference on Computational Linguistics (COLING), pages 1086-1090, Saarbrücken, Germany.
- Och, F.J. and H. Ney (2000b). Improved Statistical Alignment Models. In: Proceedings of the 38th Annual Conference of the Association for Computational Linguistics, pp. 440-447, Hongkong, China (2000)
- Och, F.J. (2001). mkcls: Training of word classes. <http://www-i6.informatik.rwth-aachen.de/Colleagues/och/software/mkcls.html>.
- Och, F.J. and H. Ney (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics 29(1) March 2003: 19-52.
- Och, F.J. (2003). GIZA++: Training of statistical translation models <http://www.isi.edu/~och/GIZA++.html>.

Petterstedt, M. (2003). I*FDG - a graphical user interface tool that enables the user to tag and convert different file formats. Natural Language Processing Laboratory (NLPLAB), Department of Computer and Information Science, Linköping University, Sweden.

Tiedemann, J. (2003). Recycling Translations - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing. Doctoral Thesis. Uppsala University. Department of Linguistics. ISSN 1652-1366, ISBN 91-554-5815-7.

Van der Eijk, P. (1993). Automating the acquisition of bilingual terminology. In Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics (EACL), pages 113-119, Utrecht/The Netherlands.

Véronis, J. and Langlais, P. (2000). Evaluation of parallel text alignment systems. the ARCADE project. In Jean Véronis, editor, Parallel Text Processing, Text, speech and language technology series, chapter 19. Kluwer Academic Publishers, Dordrecht.

Evaluation of Two Word Alignment Systems

Appendix A: Concept definitions

To understand language technology expressions, the main concepts used in the project are listed here.

Alignment	The process of selecting correspondent units in a source and target text. Alignment is here identical to linking, that is establishing links in a bitext.
Annotator	The user who is aligning segments between a source and a target text.
Bitext	An abstract concept refers to the two texts (monolingual texts) where one is the original text and the other the translation of the original. The two texts are usually named source and target text.
Discontinuous link	Link from one word or phase to several words, which are not continuous.
Evaluation	A way for the system developers to know where and what to improve in a system, for the users to be aware of strengths and weaknesses of the systems.
EGYPT	A software toolkit for corpus preprocessing, training and etc.
GIZA	A software tool in EGYPT, it is used for training translation model.
GIZA++	Extensions of GIZA. Software that will generate translation models, support up to IBM model 5 and includes many new features compared to GIZA.
Gold standard	A sample of the bitexts that has been pre-aligned manually by one or several annotators and then used to test the alignment output automatically.
Heuristics	The concept "heuristics" means commonly to solve a problem by trial and error based on experience. The heuristics in I*Link combines different methods together with resources to generate translation candidate links.
I*Eval	An evaluation tool for link results from I*Link and I*Trix.
I*Trix	An automatic tool for creating and storing associations between segments in a bitexts.
Link(s)	The result of the alignment process. Links are

	correspondent units between the source and target text.
Multi-word units (MWUs),	Word sequences and word groups
Parallel corpora	Natural language utterances and their translations with alignments between corresponding segments in different languages.
Sentence pair	A source and target language sentence that are translations of each other.
Source text	The original text which has been translated to the target text.
Statistical Ratio	Defined as Words/Unique Words in this paper.
Resource	A bilingual lexicon which is a collection of models. The resource can hold lexical information of different levels which currently are the Word form, base, POS and functional aspects.
Target text	The translation of source text.
Unit	In this manual unit refers either to a segment of a monolingual text like a word, phrase, sentence etc. but also the corresponding segments in a bitext, i.e. translation units.
Word alignment	A process of determining which words in a given source sentence should be translated to the words in a given target sentence

Appendix B: Running Giza++ at IDA

GIZA ++ is an extension of the program GIZA. Installing it in IDA, following steps need to be done.

1. Download GIZA++ source code from [here](#) to your home directory.
2. Unzip the file by using

```
gunzip GIZA++.2001-01-30.tar.gz
```

```
tar -xvf GIZA++.2001-01-30.tar
```

3. Edit file **Makefile** in GIZA++ folder.
 - a. Set the path where GIZA++ will be installed. For example:
INSTALLDIR = /home/x03xiawa/thesis/GIZA++
 - b. Set the location of the gnu compiler. For example: **CC = g++**
 - c. Set the shell that you are using. For example: **SHELL = /bin/sh**
4. Type “**make**” to make the executable program.
5. To compile plain2snt.cc by using:

```
g++ -o plain2snt.out plain2snt.cc
```

6. Download mkcls source code from [here](#).
7. Unzip the file and Edit **Makefile**:
 - a. Set the path where mkcls will be installed. For example:
INSTALLDIR = /home/x03xiawa/thesis/GIZA++/mkcls
 - b. Set the location of the gnu compiler. For example: **CC= g++**
 - c. Set the shell that you are using. For example: **SHELL = sh**
8. Create an empty file “**dependencies**” in mkcls folder by using:

```
cat > dependencies
```

then Ctrl-D

9. Type “**make**” to make the executable program.
10. Compile a bilingual corpus that is sentence aligned. For example:

Evaluation of Two Word Alignment Systems

cat *.e > english

cat *.t > swedish

11. Run plain2snt.out which is located in the GIZA++ package. In this case, Swedish is the source language and English is the target language.

plain2snt.out english swedish

Three output files will be created:

- a. english.vcb
- b. swedish.vcb
- c. englishswedish.snt

12. Run mkcls to create word classes. mkcls is a separate package.

/home/x03xiawa/thesis/GIZA++/mkcls/_mkcls -penglish -Venglish.vcb.classes

/home/x03xiawa/thesis/GIZA++/mkcls/_mkcls -pswedish -Vswedish.vcb.classes

Four output files will be created:

- a. english.vcb.classes
- b. english.vcb.classes.cats
- c. swedish.vcb.classes
- d. swedish.vcb.classes.cats

13. Run GIZA++

/home/x03xiawa/thesis/GIZA++/GIZA++ -S english.vcb -T swedish.vcb -C englishswedish.snt

Appendix C: Code for xml2text

Here is the code used to converting xml file to plaintext without numbers.

```
import java.util.*;
import java.io.*;
import org.jdom.*;
import org.jdom.input.*;
import org.jdom.output.*;
import org.jdom.xpath.*;
public class xml2txt
{
    public xml2txt(String input, String output)
    {
        try
        {
            SAXBuilder builder = new SAXBuilder();
            Document doc = builder.build(new File(input));
            FileWriter writer = new FileWriter(output);
            Element root = doc.getRootElement();
            List result =
root.getChild("text").getChild("body").getChild("div").getChild("p").getChildren();
            Iterator sentences = result.iterator();
            while(sentences.hasNext())
            {
                Element sentence = (Element)sentences.next();
                List wordList = sentence.getChildren();
                Iterator words = wordList.iterator();
                boolean first = true;
                while(words.hasNext()){
                    Element word = (Element)words.next();
                    if(first)
                        first = false;
                    else
                        writer.write(' ');
                    String text = word.getText();
                    writer.write(text,0,text.length());    }
                writer.write('\n');
            }
            writer.close();
        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
    }
    public static void main(String[] args)
    {
        new xml2txt(args[0],args[1]);
    }
}
```