University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Using statistical methods to create a bilingual dictionary

D. Hiemstra

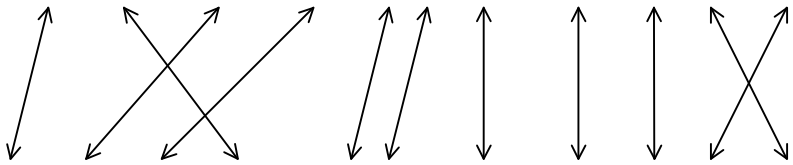
August 1996

Master's thesis
Parlevink Group
Section Software Engineering and Theoretical Informatics
Department of Computer Science

Committee:

Prof.dr. F.M.G. de Jong
Ir. W. Kraaij
Dr.ir. H.J.A. op den Akker
Dr. W.C.M. Kallenberg

(null) (null) Humanity stands at a defining moment in history (null)



De mensheid is aangekomen op een beslissend moment in haar geschiedenis

First sentence of the Agenda 21 corpus.

Abstract

A *probabilistic bilingual dictionary* assigns to each possible translation a probability measure to indicate how likely the translation is. This master's thesis covers a method to compile a probabilistic bilingual dictionary, (or bilingual lexicon), from a parallel corpus (i.e. large documents that are each others translation). Two research questions are answered in this paper. In which way can statistical methods applied to bilingual corpora be used to create the bilingual dictionary? And, what can be said about the performance of the created bilingual dictionary in a multilingual document retrieval system?

To build the dictionary, we used a statistical algorithm called the EM-algorithm. The EM-algorithm was first used to analyse parallel corpora at IBM in 1990. In this paper we took a new approach as we developed an EM-algorithm that compiles a *bi-directional* dictionary. We believe that there are two good reasons to conduct a bi-directional approach instead of a uni-directional approach. First, a bi-directional dictionary will need less space than two uni-directional dictionaries. Secondly, we believe that a bi-directional approach will lead to better estimates of the translation probabilities than the uni-directional approach. We have not yet theoretical proof that our symmetric EM-algorithm is indeed correct. However we do have preliminary results that indicate better performance of our EM-algorithm compared to the algorithm developed at IBM.

To test the performance of the dictionary in a multilingual document retrieval system, we built a document retrieval environment and compared recall and precision of a mono-lingual (Dutch) retrieval engine with recall and precision of a bilingual (Dutch-to-English) retrieval engine. We used the bilingual dictionary, compiled with the EM-algorithm, to automatically translate Dutch queries to corresponding English queries. The experiment was conducted with the help of 8 volunteers or *naive users* who formulated the queries and judged the relevance of the retrieved documents. The experiment shows that even a simple probabilistic dictionary is useful in multilingual document retrieval. With a precision of 67% and relative recall of 82%, the multilingual retrieval seems to perform even better than the monolingual Dutch system, that retrieved documents with precision of 78%, but relative recall of only 51%. There are two reasons for the good performance of the multilingual system compared to the monolingual system. First, partially correct translated queries still retrieve relatively many relevant documents because of the limitation of our domain. Secondly, linguistic phenomena in Dutch make monolingual Dutch document retrieval a more complicated problem, than monolingual English document retrieval.

Samenvatting

Een *probabilistisch tweetalig woordenboek* voegt aan elke mogelijke vertaling een mate van waarschijnlijkheid toe om aan te geven hoe aannemelijk de vertaling is. Dit afstudeerverslag beschrijft een methode waarmee een probabilistisch tweetalig woordenboek (of tweetalig lexicon) gegenereerd kan worden uit een *parallel corpus* (d.w.z uit grootte documenten die elkaars vertaling zijn). In dit verslag worden twee onderzoeksvragen beantwoord. Hoe kunnen statistische methoden toegepast op parallelle corpora worden benut om een tweetalig woordenboek te genereren? En, welke uitspraak kan worden gedaan over prestatie van het gegenereerde woordenboek in een meertalig 'retrieval' systeem.

Voor het genereren van het woordenboek is gebruik gemaakt van een statistisch algorithme genaamd het EM-algoritme. Het EM-algoritme werd voor het eerst in 1990 bij IBM gebruikt om parallel corpora the analyseren. Dit verslag beschrijft een nieuwe aanpak: het ontwerp van een EM-algoritme dat in staat is een bidirectioneel woordenboek te genereren. Voor de bidirectionele aanpak, in plaats van een mono-directionele aanpak zijn twee goede redenen te noemen. Ten eerste zal een bi-directioneel woordenboek minder ruimte innemen dan twee monodirectionele woordenboeken. Ten tweede zijn we van mening dat een bi-directionele aanpak zal leiden tot betere schattingen van de waarschijnlijkheden dan de mono-directionele aanpak. Er is nog geen theoretische onderbouwing van de correctheid van het EM-algoritme dat in dit verslag is ontwikkeld. De resultaten van het algoritme duiden er echter op dat ons EM-algoritme betere prestaties levert dan het EM-algoritme dat ontwikkeld is bij IBM.

Om de prestaties van het woordenboek op een meertalig 'retrieval' systeem te testen, is 'recall' en 'precision' van het enkeltalige (Nederlandse) 'retrieval' systeem vergeleken met 'recall' an 'precision' van het tweetalige (Nederlands/Engelse) 'retrieval' systeem. Het tweetalige woordenboek, dat we gegenereerd hebben met behulp van het EM-algoritme, is gebruikt voor het automatisch vertalen van Nederlandse zoekvragen naar de bijbehorende Engelse zoekvragen. Het experiment is uitgevoerd met de hulp van 8 proefpersonen of *naïeve gebruikers* om de zoekvragen te formuleren en de relevantie van de gevonden documenten te beoordelen. Het experiment laat zien dat zelfs een simpel probabilistisch woordenboek nuttig is in een meertalige 'retrieval' systeem. Met een 'precision' van 67% en 'relative recall' van 82%, lijkt het meertalige systeem beter te werken dan het enkeltalige systeem dat documenten vond met een 'precision' van 78%, een 'relative recall' van slechts 51%. Er zijn twee redenen te noemen voor de goede prestatie van het meertalige systeem in vergelijking met het enkeltalige systeem. Ten eerste zullen gedeeltelijk vertaalde zoekvragen relatief veel relevante documenten vinden door het beperkte domein van ons systeem. Ten tweede zorgen bepaalde taalkundige eigenschappen van het Nederlands ervoor dat 'retrieval' met Nederlands als taal gecompliceerder is dan 'retrieval' in het Engels.

Preface

Writing this thesis took the spring and summer of 1996. It gave me the opportunity to make one statement with absolute, one-hundred percent certainty: what they say about inspiration and transpiration is true. Fortunately, only a small part of the inspiration and transpiration was actually mine. However, the parts that were mine gave me more fun than I thought it would be before I began to work on this project. Credits for this largely go to Franciska de Jong for being even more chaotic than I am, Wessel Kraaij for always being critical via e-mail and Rieks op den Akker for teaching me some lessons on planning and organising. I would specially like to thank Wilbert Kallenberg, who often took a lot of time for me to make the world of Statistics an exciting one.

The following people helped me a lot by giving the retrieval system I built a hard time: Ursula Timmermans, Els Rommes, Arjan Burggraaf, Charlotte Bijron, Theo van der Geest, Anton Nijholt, Toine Andernach and Jan Schaake. I like to thank them very much, specially those volunteers that had to read more than 10 pages of dull fragments.

Writing this thesis not only concludes my terminal project, it also concludes seven years of college life. During these years, I met a lot of new friends and many of them probably wondered, one time or another, if I ever would graduate. I know I did. Still, I would like to thank every one who accompanied me during my college years. Most of all, however, I would like to thank my girlfriend Ursula, who never stopped believing in me and is now put in the right.

Djoerd Hiemstra
Enschede, August 1996

Contents

Chapter 1

Introduction 1

1.1 Multilingual document retrieval 1
1.2 The approach 1
1.3 Research questions 2
1.4 Organisation of this paper 2

Chapter 2

The Twenty-One Project 5

2.1 Objective of Twenty-One 5
2.2 Functionality of a document information system 5
2.2.1 Document maintenance 6
2.2.2 Document profiling 6
2.2.3 Fully automated document profiling 6
2.2.4 Document retrieval 7
2.3 Machine translation techniques 7
2.4 The role of Agenda 21 7

Chapter 3

Advances in Statistical Translation 9

3.1 Rationalism vs. Empiricism 9
3.2 Assigning probabilities to translations 10
3.2.1 Defining equivalence classes 10
3.2.2 The model 10
3.2.3 Finding a statistical estimator 10
3.2.4 Hypothesis testing 11
3.2.5 Discussion 11
3.2.6 About the remaining paragraphs 11
3.3 Morphology 11
3.3.1 Knowledge based analysis of morphology 11
3.3.2 Statistical analysis of morphology 12
3.3.3 Discussion 12
3.4 Sentence alignment in a bilingual corpus 12
3.4.1 Identifying words and sentences 12
3.4.2 Alignment of sentences based on lexical items 13
3.4.3 Alignment of sentences based on sentence length 13
3.4.4 Discussion 13
3.5 Basic word alignment in a bilingual corpus 14
3.5.1 Alignments 14
3.5.2 Algorithms 15
3.5.3 Results 16
3.5.4 Discussion 16
3.6 Recent research on statistical translation 16
3.6.1 Statistical translation using a human dictionary 16
3.6.2 Statisical translation using a MT lexicon 17
3.6.3 Finding noun phrase correspondances 17

3.6.4 Translation of collocations 17
3.7 Discussion 17

Chapter 4

Definition of equivalence classes 19

4.1 Introduction 19
4.2 Equivalence classes 19
4.3 Class definition problems 21
4.3.1 periods 21
4.3.2 hyphens and dashes 21
4.3.3 abbreviations and acronyms 21
4.3.4 apostrophes 21
4.3.5 diacritics 21
4.4 Contingency tables 22
4.4.1 Dividing the observations with 'complete data' 22
4.4.2 Dividing the observations with 'incomplete data' 23

Chapter 5

The translation model 25

5.1 Introduction 25
5.2 Information Theoretic approach 25
5.3 The prior probability: modelling sentences 26
5.4 The channel probability: modelling translations 26
5.5 Drawing a parallel corpus 27
5.6 Symmetry of the model 27
5.7 Discussion 28

Chapter 6

MLE from incomplete data: The EM-algorithm 29

6.1 Formal approach to MLE 29
6.2 Criticism on MLE 30
6.3 Complete data vs. incomplete data 31
6.3.1 Definition of the incomplete data 31
6.3.2 Definition of the complete data 31
6.4 Definition of the EM algorithm 32
6.5 Implementation of the E-step 32
6.5.1 Definition of the sufficient statistics 32
6.5.2 Counting down all combinations 33
6.5.3 Combining equivalence classes 33
6.5.4 Iterative proportional fitting 34
6.5.5 Brown's E-step 34
6.5.6 A dirty trick 35
6.6 Implementation of the M-step 35
6.7 Comparing the different algorithms 35
6.7.1 Combining equivalence classes 36
6.7.2 IPFP with p_{ij} as initial estimate 36
6.7.3 Brown's E-step 36
6.7.4 IPFP with 'dirty trick' initial estimate 37
6.8 Discussion 37

Chapter 7

Evaluation using Agenda 21 39

7.1 The experiment 39
7.1.1 Dividing the corpus 39
7.1.2 Training the parameters 39
7.1.3 Using the testing corpus as a document base 39

7.1.4 Measuring retrieval performance 40
7.2 The results 41
7.2.1 Global corpus characteristics 41
7.2.2 Some preliminary results 42
7.2.3 The multilingal IR results 43
7.2.4 Discussion of the multilingual IR results 44
7.2.5 Conclusion 46

Chapter 8

Conclusions 47

8.1 Building the dictionary 47
8.2 The bilingual retrieval performance 47
8.3 Recommendations to improve the translation system 48
8.3.1 Equivalence classes 49
8.3.2 The translation model 49
8.3.3 Improving the parameter estimation 49

Appendix A

Elementary probability theory ..... a

A.1 The axiomatic development of probability ..... a
A.2 Conditional probability and Bayesian Inversion ..... a
A.3 Random variables ..... b
A.4 Expectation of a random variable ..... b
A.5 Joint, marginal and conditional distributions ..... b

Appendix B

Linguistic phenomena c

B.1 Morphology c
B.1.1 Advantages of morphological analysis c
B.1.2 Areas of morphology c
B.2 Lexical ambiguity ..... d
B.3 Other ambiguity problems ..... d
B.4 Collocations and idiom ..... d
B.5 Structural differences ..... e

Appendix C

Statistical Estimators ..... g

C.1 Introduction ..... g
C.2 Maximum likelihood estimation ..... g
C.3 Laplace's Law ..... h
C.4 Held out estimation ..... h
C.5 Deleted estimation i
C.6 Good-Turing estimation i
C.7 Comparison of Statistical Estimators i

Appendix D

The Implementation ..... k

D.1 Sentence identification ..... k
D.2 Sentence alignment ..... k
D.3 Word alignment l
D.4 The retrieval engine l

Chapter 1

Introduction

The recent enormous increase in the use of networked information and on-line databases has led to more databases being available in languages other than English. In cases where the documents are only available in a foreign language, multilingual document retrieval provides access for people who are non-native speakers of the foreign language or not a speaker of the language at all. A translation system or on-line dictionary can be used to identify good documents for translation.

1.1 Multilingual document retrieval

Where can I find Hungarian legislation on alcohol? What patent applications exists for certain superconductivity ceramic compounds in Japan and which research institutes lay behind them? Is it true that the Netherlands are a drugs nation? Who was the last French athlete that won weight lifting for heavy weights? And, where can I find the text of the last German song that won the Eurovision Song Festival?

Probably the answer of most of the questions above can be found somewhere on the Internet. Those that cannot be found, probably can be within the next couple of years. However, to find the answers, we must have some knowledge of Hungarian, Japanese, Dutch, French and German. Wouldn't it be nice if the search engine we used was capable of translating our English *query* to for example Hungarian, so we can decide which Web pages we would like to have translated by a human translator?

For the purpose of multilingual document retrieval, such a search engine must have access to a bilingual (or multilingual) dictionary to translate queries (or indexes). Existing bilingual dictionaries are either too expensive or inadequate for qualitative good translation of the queries. Tackling the problem of acquiring the bilingual dictionary is the main objective of this paper.

1.2 The approach

In this paper we describe a systematic approach to build a bilingual probabilistic dictionary for the purpose of document retrieval. A *probabilistic* dictionary assigns a probability value to each possible translation in the dictionary. Our limited objective was to build a bilingual English/ Dutch retrieval system on the domain of ecology and sustainable development.

We compiled the dictionary by comparing large documents that are each others translation. A document and its translation will be called a *bilingual* or *parallel corpus* throughout this paper. To analyse the bilingual corpus we used a *statistical algorithm* called the *EM-algorithm* that was first used to analyse bilingual corpora at IBM in 1990 [Brown, 1990]. Their article inspired

research centra all over the world to use statistical methods for machine translation also. This paper contributes to this broad discussion in two new ways.

1. We developed an EM-algorithm that compiles a bi-directional dictionary (that is, a dictionary that can be used to translated from for example English to Dutch *and* Dutch to English). We believe that there are two good reasons to conduct a bi-directional approach. First, a bi-directional dictionary will need less space than two unidirectional dictionaries. The compilation of a bi-directional dictionary is a first step to a true multi-lingual application. Secondly, we believe that a bi-directional approach will lead to better estimates of the translation probabilities than the uni-directional approach.
2. We built a document retrieval environment and compared recall and precision of a mono-lingual (Dutch) retrieval engine to recall and precision of a bilingual (Dutch-to-English) retrieval engine. We used the bilingual dictionary, compiled with the EM-algorithm, to automatically translate Dutch queries to corresponding English queries. The experiment was conducted with 8 volunteers or *naive users* who formulated the queries and judged the relevance of the retrieved documents

1.3 Research questions

In the statistical methods to translate index terms of documents, the following research questions will be answered in this paper

1. In which way can statistical methods applied to bilingual corpora be used to create the bilingual dictionary?
2. what can be said about the performance of the created bilingual dictionary in a multilingual IR system?

1.4 Organisation of this paper

This paper is organised as follows:

Chapter 2, *The Twenty-One project*, situates the research questions formulated in the previous paragraph in a broader context: the Twenty-One project.

Chapter 3, *Advances in Statistical Translation*, introduces the field of statistical natural language processing and gives some of the results of previous research on the topic. Readers who are familiar with the field of Statistical Translation may want to skip this chapter. Still, on the first time through, the reader who is *not* that familiar with the topic may also wish to skip chapter 3, returning to this chapter if he or she wants to know more about the background of the research presented in the rest of this paper.

The next 3 chapters follow the three basic steps that have to be followed to compile the dictionary, if we assume that it is known which sentences in the corpus are each others translation: First, the definition of equivalence classes. Secondly, the definition of the translation model and, finally, the definition of the statistical estimator and the estimating algorithm (the EM-algorithm).

Chapter 4, *Definition of equivalence classes*, discusses some of the basic tools we need if we are going to analyse (bilingual) corpora. First the concept of equivalence classes is introduced together with possible solution of some basic class definition problems. After that we introduce the concept of contingency tables.

Chapter 5, *The translation model*, covers the application of an Information Theoretic approach to the translation of sentences. In this chapter we define simple but effective ways to model sentences and the translation of sentences.

Chapter 6, *The EM-algorithm*, discusses the definition of different EM-algorithms and some preliminary results of the different algorithms

The last two chapters cover the results of our research.

Chapter 7, *Evaluation using Agenda 21*, discusses the results of our research. First we will look at some of the entries of the dictionary we compiled to give an impression of the performance of our algorithm. After that we will give the results of the experiment we have taken to measure the usefulness of the dictionary we compiled.

Chapter 8, *Conclusions*, discusses the conclusions of this paper and gives recommendations for future research on the topic

Appendix A, *Elementary probability theory*, contains definitions of the mathematical notations used throughout the paper

Appendix B, *Linguistic phenomena*, contains definitions of linguistic phenomena we refer to in this paper. It particularly discusses *morphology*, *ambiguity*, and *collocations* / *idiom*.

Appendix C, *Statistical Estimators*, covers some different statistical estimators like Maximum Likelihood Estimation, Deleted Estimation and Good-Turing Estimation and its performance on a bigram model.

Appendix D, *Implementation of the EM-algorithm*, contains the important comments on the implementation of the sentence identification, sentence-alignment, EM-algorithm and IR environment.

Chapter 2

The Twenty-One Project

The Twenty-One project is a project that is funded by the European Union and has participants in different European countries. One of its participants is the Twente University in the Netherlands. In this chapter an overview will be given of the project parts that are the most relevant for the research problem we formulated in the previous chapter. First, we will look at the main objective of the Project Twenty-One and at the users of Twenty-One. Then we will look at the three main activities within Twenty-One: document maintenance, document profiling (or indexing) and document retrieval. Finally we will look at the role of the document Agenda 21 in the project [Gent, 1996].

2.1 Objective of Twenty-One

There are two problems that prevent effective dissemination in Europe of information on ecology and sustainable development. One is that relevant and useful multimedia documents on these subjects are not easy to trace. The second problem is that although the relevance of such documents goes beyond the scope of a region or country, they are often available in one European language only. The Twenty-One project aims at improving the distribution and use of common interest documents about ecology and sustainable development in Europe. The improvement will be achieved by developing knowledge-based document information technology and by providing the current network of European organisations with the knowledge to improve their information distribution by using this new technology.

Twenty-One aims at people and organisations that in one way or another have to deal with the development of environment preserving behaviour. The environmental organisation that will use Twenty-One act both as users and providers of information about the environment.

The main objective of Twenty-One is to develop a domain-independent technology to improve the quality of electronic and non-electronic multimedia information and make it more readily and cheaply accessible to a larger group of people.

2.2 Functionality of a document information system

Document information systems usually have three functionalities (see figure 2.1): maintenance, profiling and retrieval. [Hemels, 1994]. Each functionality has its own user interface and its own user type.

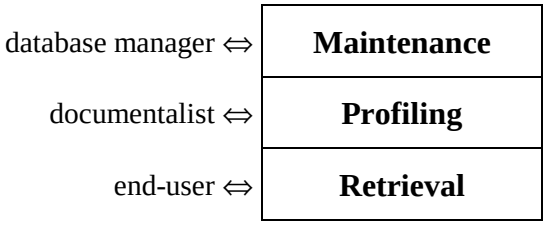


Figure 2.1, Functionality of a document information system

2.2.1 Document maintenance

Document maintenance is the set of administrative activities needed to maintain the document base. These activities include for example adding or removing documents, changing or updating them, keeping consistency, maintaining catalogue and thesaurial systems, etc. Computers can be readily used to administer these information. A staff of people with various skills is usually involved in document maintenance.

2.2.2 Document profiling

Document profiling is the process of attributing features to documents such that they can be retrieved successfully. These features may include search keys such as index keywords, classification codes or even abstracts. Four sorts of features usually are distinguished.

1. Non-contents descriptions. These are features about the position of a document within its domain, such as author, date, publisher, addressee, etc. Very often this information cannot be found in the document itself.
2. Contents descriptions. These are descriptions of the content of a document.
3. Judgements. These are features about the quality of the document related to it's use in a process.
4. Corpus selection. This is the determination of the suitability of a document for the document base.

In traditional library maintenance systems, document profiling is done manually. Therefore it's time consuming. It is also the most expensive activity within traditional library systems because it has to be done by experts (documentalists). The profiling of the contents descriptions has to be automated to reduce these costs.

2.2.3 Fully automated document profiling

Because of the costs in both time and money, fully automated profiling of the contents descriptions is one of the most important issues in the Project Twenty-One. Two processes are required. First, automated profiling of documents requires software that can differentiate text form graphics. A second process analyses the text part of the document.

Twenty-One will use *full text retrieval* (i.e. all text is used) tools to build the index terms of the documents. Most commercial full text retrieval tools only use statistics to build index terms of documents. Unlike these full text retrieval tools, Twenty-One will use two *knowledge based* techniques to built these indexes:.

1. Natural Language Processing (NLP) to find noun phrases to build document indexes consisting of phrases.
2. Knowledge based layout analysis to recognise those parts of the documents that have the highest information density, such as titles, captions, abstracts or introductions.

2.2.4 Document retrieval

Document retrieval involves the searching for documents in the document base. Document retrieval is performed by end users possibly assisted by librarians or other information specialists. In addition to the knowledge based techniques mentioned in the paragraph above, the project will develop two other techniques for document retrieval.

3. Automated hyper linking. This is the attachment of links between documents at places where terms are the same or alike. Hyper linking is the de facto standard in Internet communication.
4. Fuzzy matching. Fuzzy matching breaks down both the index and the query in primitives. Both sets of primitives can be matched by intersection.

2.3 Machine translation techniques

The machine translation problem in the project is a very special one. The translation process only has to deal with noun phrases. The process does not have to translate difficult syntactic structures. This makes translation an relatively easy task. On the other hand, the project requires domain independence. This means, in the case of Twenty-One, that the techniques used to translate the noun phrases cannot make use of the knowledge of for example sustainable development issues in Agenda 21. The project's requirements of domain-independence preclude the use of existing commercial translation software, because this type of software heavily depends on domain modelling, like translation memory, thesauri or model-theoretic semantics, or interaction with human translators.

In this paper research will be made into domain independent translation techniques that are useful in document retrieval environments like the Twenty-One environment. That is, because we will use bilingual corpora to build a dictionary, the dictionary itself will be domain-specific. However, the technique used to build the dictionary will be domain-independent and (almost) fully automatic. Existing dictionaries can be modified easily if a bilingual corpus of another domain is available.

2.4 The role of Agenda 21

Agenda 21 is an international document available in all European languages reflecting the results of the United Nations Conference 1992 on ecology in Rio de Janeiro. It contains guiding principles for sustainable development covering topics such as patterns of consumption, cultural changes, deforestation, biological diversity, etc. These topics are good examples of the phrase indexes Twenty-One has to extract from the documents and translate to indexes in other languages.

Agenda 21 is a document that is already available and accessible on a multilingual basis. The project Twenty-One however aims at the disclosure of documents that are not available on a multilingual basis, i.e. are *not* like Agenda 21. Because Agenda 21 is available in all different languages the performance of a multilingual retrieval system can easily be evaluated by comparing the results of disclosure of the official translations.

Chapter 3

Advances in Statistical Translation

As stated in chapter 1 we will use statistical methods to create the dictionary. This chapter begins by attempting to situate the statistical approach in the field of computational linguistics. After that, we will look at attempts of leading research centra like IBM and AT&T to find translations of words using large parallel corpora.

3.1 Rationalism vs. Empiricism

Between about 1960 and 1990 most of linguistics, psychology and artificial intelligence was dominated by a rationalist approach. A rationalist approach is characterised by the belief that a significant part of the knowledge in the human mind is not derived by the senses, but is fixed in advance, presumably by genetical inheritance. Arguments for the existence of an innate language faculty were introduced by Noam Chomsky [Chomsky, 1965]. Developing systems for natural language processing (NLP) using a rationalist approach leads to systems with a lot of handcoded starting knowledge and reasoning mechanisms.

In contrast an empiricist approach argues that knowledge derives from sensory input and a few elementary operations of association and generalisation. Empiricism was dominant in most of the fields mentioned above between 1920 and 1960, and is now seeing a resurgence in the 1990s. An empiricist approach to language suggests that we can learn the structure of language by looking at large amounts of it. Developing systems for NLP using an empiricist approach leads to relatively small systems that have to be trained using large corpora.

Claude Shannon who single handedly developed the field of information theory [Shannon, 1948] can be seen as the godfather of the modern statistical NLP. In fact Chomsky referred mainly to Shannon's *n*-gram approximations [Shannon, 1951] as he introduced his famous 'colorless green ideas'. Chomsky's criticism of *n*-grams in *Syntactic Structures* [Chomsky, 1957] ushered in the rationalist period. The most immediate reason for the renaissance of empiricism in the 1990s is the availability of computers which are many orders of magnitude faster than the computer in the 1950s. Large machine-readable corpora are now readily available.

The debate described above is also found in the philosophy of many other fields of science. In fact Plato argued about 2,400 years ago that our knowledge about truth, beauty and honesty is already present when we are born. Not much later his student, Aristotle, wrote that ideas were not native: We learn what is beautiful in life because we learn from our parents and because they reward certain opinions. In NLP the debate still continues, whether it is characterised in terms of *empirical* vs. *rationalist*, *statistics-based* vs. *rule-based*, *performance* vs. *competence*, or simply *Shannon-inspired* vs. *Chomsky-inspired*. The approach presented in this paper, follows Shannon's ideas on NLP.

3.2 Assigning probabilities to translations

In this paper, we will consider the translation of individual sentences. We take the view that every sentence in one language is a possible translation of any sentence in the other. We assign to every pair of sentences (E,D) a probability $P(E|D)$ to be interpreted as the probability that a translator will produce E in the target language when presented with D in the source language. We expect the probability $P(E|D)$ to be very small for pairs like (*I'm a twenty-first century digital boy, De mensheid is aangekomen op een beslissend moment in haar geschiedenis*) and relatively large for pairs like (*Humanity stands at a defining moment in history, De mensheid is aangekomen op een beslissend moment in haar geschiedenis*). More about probability theory can be found in appendix A.

How do we determine the value of the probability measure P when we are dealing with the translation of sentences? Determining P takes three basic steps.

1. dividing the training data in equivalence classes; each class will be assigned a probability parameter;
2. determining how to model the observations;
3. finding a good statistical estimator for each equivalence class parameter / test a hypotheses about the unknown parameters

The last step leaves us with two possibilities: *estimation* or *hypothesis testing*. Let us consider a simple linguistic (monolingual) problem: What is the probability of selecting the word *sustainability* if someone is randomly selecting 100 words from the Agenda 21 corpus.

3.2.1 Defining equivalence classes

We define two equivalence classes ω_1 and ω_2 which are assigned the parameters p_1 en p_2. If we observe the word *sustainability* then the observation is contributed to ω_1, if we observe any other word the observation is contributed to ω_2. Because the sum of the probability over all possible events must be one there is only one unknown parameter. The other parameter is determined by $p_2 = 1 - p_1$.

3.2.2 The model

The probability measure P is unknown and probably very complex. However, a satisfying *model* of P may be a binomial distribution [Church, 1993]. The number of times *sustainability* appears in Agenda 21 can be used to fit the model to the data. The classical example of a binomial process is coin tossing. We can think of a series of words in English text as analogous to tosses of a biased coin that comes up heads with probability p_1 and tails with probability $p_2 = 1 - p_1$. The coin is heads if the word is *sustainability* and the coin is tails if the word is not. If the word *sustainability* appears with probability p_1. Then the probability that it will appear exactly x times in an English text of n words (n tosses with the coin) is

$$P(X = x) = \binom{n}{x} p_1^x (1 - p_1)^{n-x}, \quad 0 \le x \le n \tag{1}$$

3.2.3 Finding a statistical estimator

The expected value of the binomial distributed variable X is $E(X) = np_1$. Of course, the value of p_1 of the binomial distributed word *sustainability* is unknown. However, in a sample of n words we should expect to find about np_1 occurrences of *sustainability*. There are 41 occurrences of *sustainability* in the Agenda 21 corpus, for which n is approximately 150,000. Therefore we can argue that $150{,}000p_1$ must be about 41 and we can make an estimate $\hat{p}$ of p_1 equal to

41/150,000. If we really believe that words in English text come up like heads when we flip a biased coin, then $\hat{p}$ is the value of p_1 that makes the Agenda 21 corpus as probable as possible. Therefore, this statistical estimation is called the *maximum likelihood estimation*.

3.2.4 Hypothesis testing

Instead of estimating the unknown parameter p_1 of the binomial, we also may want to test some hypothesis of the unknown parameter p_1. We can for example test the null hypothesis $H_0 : p_1 = 0.5$ against the alternative hypothesis $H_1 : p_1 \neq 0.5$. The probability of falsely accepting the alternative hypothesis can be made arbitrarily small.

3.2.5 Discussion

The example we just presented implies that we have to observe translation pairs of words to learn something about the translation of words. If we, however, look at a bilingual corpus without any knowledge of the languages, we can not possibly know which words form translation pairs. Here lies the challenge of compiling the dictionary. Before we can estimate probabilities of translation pairs, we have to find the most likely translation of each word. For example, if we observe the sentence pair (*I wait*, *ik wacht*) the translation of the English word *I* may as well be the Dutch word *ik* as the Dutch word *wacht*. Before we can estimate the probability of the translation word pair (*I*, *ik*) we have to reconstruct what actually happened. The EM-algorithm we define in chapter 6 puts both the reconstruction task and the estimating task together in an iterative algorithm.

3.2.6 About the remaining paragraphs

The remaining paragraphs of this chapter contain attempts of leading research centra like IBM and AT&T to find translations of words using large parallel corpora. We will first look at the process of morphological analysis, which may be an useful step if we are defining equivalence classes. After that we will look at the problem of aligning the sentences. The sentence alignment problem is not subject of the research in this paper and we will use the implementation of Gale and Church described in paragraph 3.4.3. After the sentence alignment problem we will look at the important task of word alignment. Finally we will look at some recent research in Statistical Translation. The accent in the remaining paragraphs lies upon the methods that can be used and the success of these methods.

3.3 Morphology

Morphology is concerned with internal structure of words and with how words can be analysed and generated. One way to obtain normalised forms of words is to employ a morphological analyser for both languages exploiting knowledge about morphological phenomena. However, there also is a second way to obtain normalised forms of words: using statistics to analyse the morphology of words in the corpus. The reader who wants to know more about morphology can find a detailed description in appendix B.

3.3.1 Knowledge based analysis of morphology

In many respects the morphological systems of the languages involved in Twenty-One are well understood and systematically documented. Nevertheless, the computational implementation of morphological analysis is not entirely straightforward. Porter's algorithm for suffix stripping is a well known, relatively simple algorithm to obtain normalised forms [Porter, 1980]. Porter's algorithm does not use linguistic information about the stem it produces. A necessary component for such analysis is a dictionary, slowing down the algorithms efficiency. Porter's

algorithm is implemented for many languages, including a Dutch version by Kraaij and Pohlmann at the *Utrecht University*, which also handles prefixes, affixes, changes in syllables and duplicate vowel patterns [Kraaij, 1994].

3.3.2 Statistical analysis of morphology

Another way to handle morphological phenomena is to make use of the corpus. By comparing sub strings of words, it is possible to hypothesise both stem and suffixes. This method was used by Kay and Röscheisen for their method of sentence alignment [Kay, 1993] and by Gale and Church in word-alignment [Gale, 1991].
Kay and Röscheisen looked for evidence that both stem and suffix exist. Consider for example the word *wanting* and suppose the possibility is considered to break the word before the fifth character '*i*'. For this to be desirable, there must be other words in the text, such as *wants* and *wanted* that share the first four characters. Conversely, there must be more words ending with the last three characters: '*ing*'. This method is not very accurate and overlooks morphological phenomena like changes in syllables and vowels.
Gale and Church hypothesise that two words are morphologically related if they share the first five characters. They checked this hypotheses by checking if the word is significantly often used in sentences that are aligned with the translation of the possibly morphological related word. More recent work is done by Croft and Xu [Croft, 1995].

3.3.3 Discussion

The statistical approach is more independent of the language than the knowledge based approach. However, because morphological phenomena are well understood, not much research has been made into statistical analysis of morphology. If morphological analysis is going to be used in the analysis of bilingual corpora, the knowledge-based method is preferred to the statistical based method.
Morphological analysis is often used for document retrieval purposes. In Twenty-One a *fuzzy matching* method is proposed in favour of morphological analysis. Fuzzy matching is a third option that could be considered to analyse the bilingual corpus.

3.4 Sentence alignment in a bilingual corpus

Another useful first step in the study of bilingual corpora is the sentence alignment task. The objective is to identify correspondences between sentences in one language and sentences in the other language of the bilingual corpus. This task is a first step towards the more ambitious task of finding words which correspond to each other.

There are several publications about sentence alignment. The approach of Kay and Röscheisen at Xerox [Kay, 1993] has a lexical basis, which differs considerably from the sentence length basis used in the approaches of Brown, Lai and Mercer at IBM [Brown, 1991] and Gale and Church at AT&T [Gale, 1993].

3.4.1 Identifying words and sentences

Identifying sentences is not as easy as it might appear. It would be easy if periods always were used to mark sentence boundaries, but unfortunately many periods have other purposes. They may appear for example in numerical expressions and abbreviations. A simple set of heuristics can be used to identify sentence boundaries (see paragraph 6.2). For some languages like Chinese even the identification of words is not a trivial task, because written Chinese consists of a character stream with no space seperators between words [Wu, 1995].

3.4.2 Alignment of sentences based on lexical items

This method for aligning sentences rests on being able to identify one-to-one associations between certain words, notably technical terms and proper names. The method makes an initial guess of the most probable alignment of the sentences of both texts of the corpus using a Gaussian distribution. It pairs the first and last sentences of the two texts with a small number of sentences from the beginning and end of the other text. The closer a sentence is to the middle of the text, the larger the set of sentences in the other text that are possible correspondences for it. The next step is to hypothesise a set of words that are assumed to correspond based on the similarities between their distributions in the two texts. With this hypotheses a new guess can be made of the alignments of the sentences. The method converges in about four of these steps.

Gale and Church tested their method on two pairs of articles form *Scientific American* and their German translations in *Spektrum der Wissenschaft*. They claim their method aligns 99.7% of the sentences correct, covering 96% of the sentences. The 4% that is not covered are mostly due to German subheadings not appearing in the English version. As a secondary result, the method produces a set of aligned words (technical terms and proper names) of which more than 95% is correct, and even more may be useful for information retrieval purposes. Aligning words, however, was not the primary interest of the authors. Aligning 469 English to 462 German sentences took the method about 3.5 minutes of processing time. The computational complexity of the algorithm is bound by $O(n\sqrt{n})$, with n the number of sentences [Kay, 1993].

3.4.3 Alignment of sentences based on sentence length

This method for aligning sentences is based on a very simple statistical model of character or word lengths of sentences and paragraphs. The method is a two-step process. First paragraphs are aligned, and next sentences within each paragraph. The model makes use of the fact that longer sentences in one language tend to be translated into longer sentences in the other language, and shorter sentences tend to be translated into shorter sentences. A probabilistic score is assigned to each pair of proposed sentence pairs, based on the ratio of lengths of the two sentences and the variance of this ratio.

An evaluation was performed based on a trilingual corpus of fifteen economic reports issued by the *Union Bank of Switzerland* in English, French and German. The corpus covers 725 sentences in English and a corresponding number of sentences in the other two languages. The method correctly aligned all but 4.2% of the sentences. However, by selecting the best scoring 80% of the alignments, the error rate is reduced form 4.2% to 0.7%. At IBM aligning nearly 3 million pairs of sentences from Hansard materials took 10 days of running time. The computational complexity of the algorithm is bound by $O(n)$, with n the number of sentences [Gale, 1993].

3.4.4 Discussion

Both described methods are fairly language independent and both methods are able to produce correct aligned sentences.

The lexical approach is, compared to the sentence length approach, more elegant and more robust as it aligns almost all sentences correct. As a result of it's method, the lexical approach also produces translations of technical terms and proper names. This may make the creation of the dictionary an easier job. However, each word only can have one possible translation (ambiguous words are not spotted) and one word cannot generate more words in the target language.

The sentence length approach is a much simpler and faster method than the lexical approach. By skipping alignments with a low probabilistic score, the method produces aligned sentences

that are almost as accurate as the more complicated lexical approach. The sentence length approach has the additional advantage that it's very well documented.

In later publications like [Chen, 1993] and [Wu, 1995] both methods are integrated to gain the robustness from the lexical approach and the performance of the sentence length approach.

3.5 Basic word alignment in a bilingual corpus

Now that the sentences in the corpus are identified and aligned, a probabilistic dictionary can be generated by aligning the words also. The idea of an alignment between a pair of strings was introduced by [Brown, 1990]. Again, the same research centra as mentioned above have made advances in the word-alignment problem. This time [Brown, 1993] at IBM show a different approach from the approach taken by [Gale, 1991] at AT&T. First in this paragraph, three different types of alignment will be distinguished. Then, different algorithms for finding the most probable alignment will be described. Finally the successes of both approaches are discussed.

3.5.1 Alignments

If word correspondences have to be found in sentences, a alignment model has to describe how source language words can be translated into target language words. First it is decided which type of alignment is allowed. After that it must be decided which parameters describe the probability of an alignment.

Unlike the alignment of sentences, with the alignment of words order constraints need not to be preserved and crossing dependencies are permitted (*a nice man* $\rightarrow$ *un homme gentil*). Different alignment models can be chosen, determining which type of alignment are allowed. The most simple model allows one to one associations. This model cannot account for the translation of most sentences. This may, however, not be necessary for compiling a simple dictionary. In a general alignment *n* words can be connected to *m* words of the other language. The following four types are distinguished by [Brown, 1993].

1. alignment with independent source and target language words,
2. alignment with independent source language words,
3. alignment with independent target language words,
4. a general alignment.

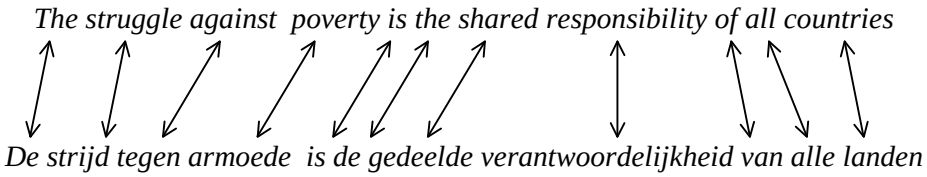


Figure 5.1, alignment with independent Dutch and English words

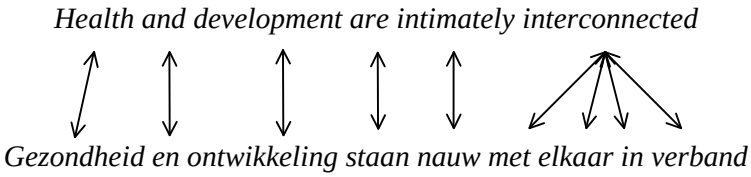


Figure 5.2, alignment with independent English words

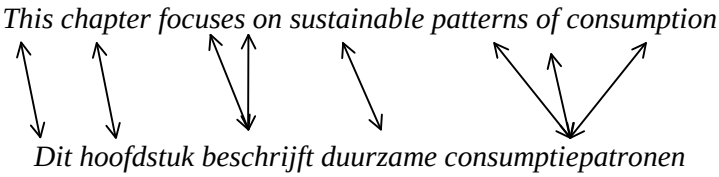


Figure 5.3, alignment with independent Dutch words

The alignment model used by [Gale, 1991] is a model of type 1. This simple model uses only translation parameters for each possible pair of a target language and source language word. The model used by [Brown, 1990] is of type 2. The model is quite complicated, describing for example parameters for fertilities of a source language word, and parameters for positions in the sentences.

3.5.2 Algorithms

The parameters of the models like the ones mentioned above have to be estimated using the parallel corpus. A useful dictionary may need several ten thousands of entries, leaving 100 million parameters to be estimated. The very large number of parameters causes many problems. It may no longer be possible to compute every parameter in the model and it may not be possible to hold a copy of every parameter in memory.

Again two very different approaches can be taken. The first approach is called *hypothesis testing*. It hypothesises some alignments in the corpus and removes all the alignments it probably accounts for from the corpus. Then other hypothesis can be made. The second approach is called *fitting models* technique or simply *estimating*. It estimates the parameters of the model several times. Each new estimation must be more probable then the former.

hypothesis testing

The first approach was taken by [Gale, 1991]. A χ^2-like statistic was used to decide which words are most likely to correspond. At some stages in the process, more and more sentences of the corpus are used to suggest possibly interesting pairs. Because of this progressive deepening strategy it is not necessary to hold a copy of every parameter in memory.

estimating

The second approach was taken by [Brown, 1990]. They used a maximum likelihood method called the Expectation Maximisation (EM-)algorithm. Each iteration of the EM-algorithm involves two steps. The first step is the expectation step. The second step is the maximisation step. These two steps are repeated several times until the state of parameters of the model don't change significantly. Sparse matrix structures were implemented to reduce the number of parameters in memory. Still their approach needs a lot of memory.

3.5.3 Results

With the hypothesis testing approach Gale et al. were able to find 6,419 translation pairs from 220,000 parallel sentences of the Hansard corpus. Based on a sample of 1,000 pairs, about 98% of the selected pairs of words were translations. With hypothesis on morphology described above they were able to find 7,047 additional translation pairs with accuracy of 98%. They suggested corresponances for about 60% of the words in the corpus, but probably a much smaller percentage of the number of different words.

With the estimating approach Brown et al. were able to align 1,778,620 parallel sentences of the Hansard corpus. For every word (42,005 different English words and 58,016 different French words) in the corpus a translation is found. The ambitious objective of Brown et al. is to develope a statistical MT system. Their results are really inspiring as the algorithm finds for example (*marquées d'un asterisque* | *starred*) and (*ne...pas*| *not*) [Brown, 1993]. In an earlier version of their system they trained the parameters on 40,000 pairs of sentences. On 73 new French sentences from elsewhere in the Hansards they were able to translate 48% to correct English [Brown, 1990].

3.5.4 Discussion

Compared to the estimating approach, the hypothesis testing approach, has the advantage that it is a very simple algorithm. It does not consider all possible translations and will therefore need less processing time and memory. On each hypothesis the probability a wrong assumption is made can be made as small as nessecary. However, here lies the big problem of this approach. If we make the probability of accepting a false translation too small, the algorithm will reject almost all hypothesises. If we make the probability too big the algorithm will 'collaps' by its own mistakes. Suppose we choose the probability of accepting a false translation to be 1%. Then we should expect to find on every 100 translations 1 false translation. Because the algorithm removes translation pairs from the corpus once a correlation between them is clearly established, the corpus will become noisier at every iteration. At some stage in the process, the corpus will be too impoverished to find any correct translation.

The estimating approach does not have this problem. The algorithm used at IBM is able to find translation pairs that a hypothesis test overlooks. In this paper we will take the estimating approach, because it is more robust and because it is able to allign all words with its most probable translation.

3.6 Recent research on statistical translation

Basic research at IBM and AT&T described in the previous paragraph inspired a lot of research centra to use statistical methods for machine translation also. In this paragraph we will look at some approaches

3.6.1 Statistical translation using a human dictionary

Research into the automatic analysis of machine readable human dictionaries was initiated by Judith Klavans and Evelyne Tzoukermann [Klavans, 1990, 1995]. They used both a parallel corpus and a human dictionary to build a bilingual lexical database called the BICORD (Bilingual Corpus-based Dictionary) system. The lexical database consists of the translation and contexts already present in the human dictionary, together with frequency counts, new translations and new contexts from the parallel corpus.

The BICORD system can be used in two complementary ways: to enhance machine readable dictionaries with statistical data and, conversely, to enhance a statistical translation system with data from the human dictionary. Statistical techniques for finding word correspondences not included in the human dictionary simple count and hypothesis testing techniques are used, rather than estimation techniques based on complex translation models. Human dictionaries were used for English to Japanese translations by Utsuro et al. [Utsuro, 1994].

3.6.2 Statisical translation using a MT lexicon

At Toshiba in Japan, Akria Kumano and Hideki Hirakawa [Kumano, 1994] genereted a MT dictionary from parallel Japanese and English texts. The method proposed utilizes linguistic information in a existing MT bilingual dictionary as well as statistical information, namely word frequency, to estimate the English translation. Over 70% accurate translations for compound nound are obtained as the first candidate from about 300 sentences Japanese/English parallel texts containing severe distortions in sentence lengths. The accuracy of the first translation candidates for unknown words, which cannot be obtained by the linguistic method is over 50%.

3.6.3 Finding noun phrase correspondances

In this algorithm noun phrase candidates are extracted from tagged parallel texts using a noun phrase recogniser. The correspondences of these noun phrases are calculated based on the EM algorithm. A sample of the Hansards comprising 2,600 aligned sentences was used to estimate the parameters. 4,900 distinct English noun phrases and distinct 5,100 French noun phrases were extracted. Accuracy of around 90% has been attained for the 100 highest ranking correspondences Evaluation has not been completed for the remaining correspondances [Kupiec, 1993].

Van der Eijk [v/d Eijk, 1993] uses a similar approach. His work differs as he uses the hypothesis testing approach instead of the estimating approach. His evaluation shows 68% accuracy. Probably, the lower accuracy is due in part to the fact that van der Eijk evaluated all translations produced by his program, while Kupiec only evaluated the top 2%. Both programs partially align each sentence with a general alignment model.

3.6.4 Translation of collocations

Frank Smadja developed a system called Champillion that identifies collocations in the source text and matches these collocations on the target text. This also includes flexible collocations that involve words separated by an arbitrary number of other words. A correlation measure called the Dice coefficient was used to measure the probability of a correspondance between the collocation and some sequence of target language word. Champillion uses a heuristic filtering stage in which to reduce the number of candidate translations. Testing Champillion on three years worth of the Hansards corpus yielded the French translations of 300 collocations for each year. About 73% of the translations was reported to be accurate [Smadja, 1996]. The program partially aligns sentences with a general alignment.

3.7 Discussion

In paragraph 3.2 we introduced the unknown probability measure P that assigns a probability $P(E|D)$ to be interpreted as the probability that a translator will produce the sentence E in the target language when presented with the sentence D in the source language. If we observe a lot of sentences, we might be able to learn something about P. We have seen that simple but effective procedures are designed to align a bilingual corpus on the sentence level. In the rest of

this paper we will assume that our bilingual corpus is indeed aligned at the sentence level. Therefore we are able to observe pairs of sentences that are each others translation.

To define the probability measure *P* we first have assign each observation to an equivalence class. Morphological analysis may be a usefull tool. It gives the process of constructing a dictionary the possibility to find statistical regularities that a full word based approach must overlook.
After we have defined the equivalence classes we have to construct a model of the translation problem. An important question is the type of word-alignment that is allowed.
Finally we have to construct an algorithm that is able assign probability values to pairs of words. Two different approaches can be taken: the hypothesis testing approach and the estimating approach. In this paper we will take the estimating approach, because it is more robust and because it is able to allign all words with its most probable translation.

Chapter 4

Definition of equivalence classes

Using the experience of previous research on the topic, we have decided to take the *parameter estimation* approach instead of the *hypothesis testing* approach. In this chapter will examine the first step of the three basic steps mentioned in paragraph 3.2 more thoroughly: We will define the equivalence classes. After that will introduce a convenient way to display the data using the equivalence classes.

4.1 Introduction

In this paragraph we are going to make the first step of the creation of a probabilistic dictionary. A probabilistic English-to-Dutch dictionary will have English entries and for each entry a list of possible translations, just like an ordinary human dictionary. For each possible translation however, a probabilistic dictionary will also give a measure of the probability of that translation.

sustainable	
duurzame	0.80
duurzaam	0.20

Figure 4.1, an example entry of a probabilistic dictionary

From the example entry of figure 4.1 we know that there are two possible Dutch translations of the English word *sustainable*. If we have to translate *sustainability* 10 times, then we should expect that it is translated 8 times to *duurzame* and 2 times to *duurzaam*. To built this dictionary we have to find translation pairs of English-Dutch words (*E*,*D*) and each pair will be assigned a probability $P(D|E)$. To define P we have to take 3 basic steps (see paragraph 3.2): defining equivalence classes, defining a translation model and defining a statistical estimator and estimating procedure for P. In this chapter we are going to look at the first step: Defining the equivalence classes.

4.2 Equivalence classes

The problem of modelling the translation of sentences is very much like problems in medicine and social sciences. In much of these studies a population of people is categorised in for example, whether a smoker or not, and different types of cancer. Frequently the physician collecting such data is interested in the relationships or associations between pairs of such categorical data.

We will do something like that in this paragraph. Suppose we want to study the bilingual corpus of figure 4.2 that consists of nine pairs of English and Dutch sentences which are each others translation. We assume that the corpus consist of randomly drawn samples of English-Dutch translations.

ik at	*I ate*
jij wacht	*you wait*
hij kan	*he can*
ik wacht	*I wait*
jij kunt	*you can*
hij at	*he ate*
ik kan	*I can*
jij at	*you ate*
hij wacht	*He waits*

Figure 4.2, An example corpus

Just like the physician has to diagnose the condition of the patient he examines ("what type of cancer does this patient have?"), we have to assign an equivalence class to every word we observe. If we perform some sort of morphological analysis we might assign the words *wait* and *waits* to the same equivalence class. Between words that fall into the same equivalence class exists an equivalence *relation*, i.e. the words share a certain property. In our example the words *wait* and *waits* share the same meaning. Often instead of talking about equivalence classes, we just talk about *categories*.

We will assume that every different word is assigned to a separate equivalence class, so for example morphological related words like *wait* and *waits* are treated as two (entirely) different words. We will however not make case-distinction of letters. So, for example the words *he* and *He* are assigned to the same equivalence class, which we will denote by *he*. There are seven different English words and also seven different Dutch words. We will define two sample spaces Ω_E and Ω_D to be

$$\Omega_E = \{I, you, he, ate, wait, waits, can\}$$
$$\Omega_D = \{ik, jij, hij, at, wacht, kan, kunt\} \quad (1)$$

Together with these sample spaces we define two probability functions $P(E_k)$ and $P(D_k)$ on the two sample spaces Ω_E and Ω_D. The events E_k and D_k are respectively the kth English and Dutch words from the sentences $E = \{E_1, E_2,...,E_l\}$ and $D = \{D_1, D_2,...,D_l\}$. However, we are not that interested in the observation of words in just the separate languages. Let us look again at the physician. The patient who's condition he has already diagnosed is asked if he or she is a smoker or not. Because the physician is interested in *relations* between types of cancer and the habit of smoking, he or she has to double the number of equivalence classes to ("cancer type 1 and a smoker", "cancer type 1 and not a smoker", "cancer type 2 and a smoker", etc.).

To model the corpus of figure 4.2, we will now form a total number of $7 \times 7 = 49$ equivalence classes, each class containing a possible English- Dutch translation. The collection of equivalence classes can be seen as a new sample space Ω on which the joint measure $P(E_k,D_k)$ of pairs (E_k,D_k) is defined. On this sample space $P(E_k)$ and $P(D_k)$ are the marginal probability distributions. The final goal is to estimate the 49 possible values p_{ij} of the joint probability measure $P(E_k,D_k)$.

4.3 Class definition problems

Splitting up a words in a text into equivalence classes or *tokens* is not as simple as it might seem at first glance. To get a flavour of what is meant we will distinct some of the problems that may arise together with some examples [Krenn, 1996].

4.3.1 periods

Like said before in chapter 3, sentence boundaries are not obvious. Periods sometimes can be part of tokens. Periods might for example be part of abbreviations, date, ordinals or enumerations. On the other hand sentence boundaries are sometimes marked by for example question marks or explanation marks.

4.3.2 hyphens and dashes

Dashes might be token internal (for example to break up a word in two parts at the end of a line) or used as punctuation (for example: *..will contain 16 processor - twice as many as...*). The distinction seems to be easy, but what to do with compounds like for example *above-mentioned* or something what is often used in Dutch, for example *laag-, middelhoog- en hoog-radioactief*?(i.e. *"low- semihigh- and high-radio active"*). Hyphens also appear in clitics such as in French verb subject inversions, for example *a-t-il* (i.e. *"has-t-he"*)
Standard solutions are not available and in most systems not much time is spend on defining the equivalence classes. Often there is a pre-processing stage in which some heuristic procedure replaces most common clitics and 'strange' compounds (like the Dutch example) by their full separate words.

4.3.3 abbreviations and acronyms

Acronyms are abbreviations that function as names or have become names. Abbreviations and acronyms appear in different shapes: Capital letters only like *UN* (i.e. *United Nations*), lowercase with periods like *i.e.* (*inter alia*), or even mixed like *Ges.m.b.H.* (i.e. *Gesellschaft mit beschränkter Haftung*). It is often hard to distinguish acronyms from ordinary words in capital letters as often appears in headlines. On the other hand it is often hard to distinguish the periods in acronyms and abbreviations from sentence ends. Often a pre-processing stage is used to replace abbreviations and/or acronyms by their full words.

4.3.4 apostrophes

There is a number of words occurring with a single quote, such as *can't*, *father's* or *agenda's* (i.e. Dutch for *agendas*). Sometimes the word can obviously be rewritten in two single words, like *can't* can be rewritten as *can not* (Note that *can not* also can be written like a compound: *cannot*). Sometime it has to be treated as a single word, like the Dutch *agenda's*) Sometimes it is only clear from context what is meant (e.g. *father's gone* vs. *father's car*, with the former a shorthand for the verb *is* and the latter a morphological marker for case genitive). Again simple 'find and replace' heuristics are often used in a pre-processing stage to resolve from shorthands.

4.3.5 diacritics

In many languages other than English, diacritics are a integral part of the spelling of words. In Dutch diacritics can be used as part of the official spelling, like *officiële* (i.e. *official*). Diacritics can also used to denote as a stress mark, like *vóór hem* or *voor hém* (i.e. *in front of him* vs. *for him*!) which is not part of the spelling of a word, but may be used if the text would be confusing otherwise. It is hard to distinguish between both uses of diacritics.

4.4 Contingency tables

We are now able to assign each observation in our example corpus to one of the 49 equivalence classes. Often these data are displayed in a table called a *contingency table*. Looking at the contingency table, each row *i* is a list of possible Dutch translations of the English word of that row, and each column *j* is a list of possible English translations of the Dutch word of that column. The contingency table is a representation that is very close to the dictionary we are going to build. The difference between the contingency table and the dictionary we are going to construct is that the contingency table contains frequency counts. The dictionary contains probabilities (and may therefore be called a probability table). The process of deriving probabilities from frequency counts is called *estimating*.

	ik	*jij*	*hij*	*at*	*wacht*	*kan*	*kunt*	
I	n_{11}	n_{12}	. .	. .	. .	. .	n_{17}	$n_{1.}$
you	n_{21}						:	$n_{2.}$
he	:						:	$n_{3.}$
ate	:						:	$n_{4.}$
wait	:						:	$n_{5.}$
waits	:						:	$n_{6.}$
can	n_{71}	. .	. .	. .	. .	. .	n_{77}	$n_{7.}$
	$n_{.1}$	$n_{.2}$	$n_{.3}$	$n_{.4}$	$n_{.5}$	$n_{.6}$	$n_{.7}$	$n_{..} = N$

Table 4.3, An example contingency table

Table 4.3 represents the general form of a contingency table using the equivalence classes defined in the previous paragraph. Here a sample of N observations is classified with respect to two qualitative variables E_i and D_j. The observed frequency or count in the ith category of the row variable E_i and the jth category of the column variable D, that is the frequency of the ijth cell of the table, is represented by n_{ij}. The total number of observations in the ith category of the row variable E is denoted by $n_{i.}$ and the total number of observations in the jth category of the column variable D is denoted by $n_{.j}$. These are known as marginal totals. In terms of the cell frequencies n_{ij} the marginal totals are given by

$$n_{i.} = \sum_{j=1}^{7} n_{ij}\,, \quad n_{.j} = \sum_{i=1}^{7} n_{ij} \tag{2}$$

The total number of observations in the sample is denoted by $n_{..}$ or simply by N. The notation of the marginal totals is known as *dot notation*, the dots indicating summation over particular subscripts [Everitt, 1992].

4.4.1 Dividing the observations with 'complete data'

Suppose our corpus exists of N = 18 observations of pairs of words which are each other translation; so, suppose we already know from the observation (*you can, jij kunt*) that *"jij"* is a translation of *"you"* and *"kunt"* is a translation of *"can"*. The data are said to be complete. We can display the observations made in the following contingency table (for convenience, the cells are reordered and only the rows and columns with totals other than 0 are displayed).

	jij	*kunt*	*hij ... at*	
you	1	0	- -	*1*
can	0	1	- -	*1*
he ...	-	-	- -	*0 ...*
waits	-	-	- -	*0*
	1	*1*	*0 ... 0*	*2*

Table 4.4, Complete observation of (you can, jij kunt)

We can now place each of the 18 observations in a table and 'add' all these tables to fill contingency table 4.3. The filled contingency table 4.3 after observing the corpus of figure 4.2 is displayed in table 4.5. The empty cells in table 4.5 have the value $n_{ij} = 0$.

	ik	*jij*	*hij*	*at*	*wacht*	*kan*	*kunt*	
I	3							*3*
you		3						*3*
he			3					*3*
ate				3				*3*
wait					2			*2*
waits					1			*1*
can						2	1	*3*
	3	*3*	*3*	*3*	*3*	*2*	*1*	*18*

Table 4.5, An example contingency table after 18 complete observations

We have not yet determined how each observation is related to the equivalence classes, so we cannot estimate the parameters p_{ij}. If we however suppose that the maximum likelihood occurs when $p_{ij} = n_{ij}/N$, then it is possible to estimate the parameters using the maximum likelihood method. For example the joint and marginal probabilities can be estimated as

$$P(can, kunt) = {}^{1}/_{18} = 0{,}0556$$
$$P(can) = P(can, ik) + P(can, jij) + P(can, hij) + ... + P(can, kunt) = {}^{3}/_{18} = 0{,}167 \quad (3)$$

And the conditional probability:

$$P(kunt \mid can) = P(kunt, can) / P(can) = {}^{0.0556}/_{0.167} = 0.333 \quad (4)$$

4.4.2 Dividing the observations with 'incomplete data'

Now, suppose our corpus consists of $N = 9$ observations of sentences which are each others translation; so, we do not know from the observation (*you can, jij kunt*) that *"jij"* is a translation of *"you"* and *"kunt"* is a translation of *"can"*. This assumption is more realistic than the one made in the previous paragraph and the data are said to be incomplete. If we try again to fill a table like table 4.4 we can only fill the marginal totals $n_{i.}$ and $n_{.j}$. The counts n_{ij} are unknown.

	jij	*kunt*	*hij ... at*	
you	**?**	**?**	- -	*1*
can	**?**	**?**	- -	*1*
he....	-	-	- -	*0 ...*
waits	-	-	- -	*0*
	1	*1*	*0 ... 0*	*2*

Table 4.6, Incomplete observation of (you can, jij kunt)

Given the marginal totals $n_{i.}$ and $n_{.j}$ and probability parameters p_{ij} it is possible to compute the expected values of the counts n_{ij}, i.e. $E(n_{ij} \mid n_{i.}, n_{.j}, p_{ij})$. It seems we have a serious problem here. Without the proper division of the observations over the equivalence classes we cannot estimate the probability parameters. Without the parameters we cannot properly divide the observation over the equivalence classes. If we have no knowledge whatsoever of p_{ij} (so knowledge of the languages we are modelling), the best thing to do seems to divide the counts for the observation (*you can, jij kunt*) equally among the n_{ij} cells.
Note that even though the possible values of the unknown n_{ij} are 0 or 1, it is possible that the expected value is a value somewhere in the interval [0,1]. If we for example toss a coin three times the expected number of heads is 1.5, even though this event cannot happen in real life.

	jij	*kunt*	*hij ... at*	
you	0.5	0.5	- -	*1*
can	0.5	0.5	- -	*1*
he....	-	-	- -	*0 ...*
waits	-	-	- -	*0*
	1	*1*	*0 ... 0*	*2*

Table 4.7, Expected complete observation of (you can, jij kunt) with no prior knowledge

We can again place each of the 9 observations in a contingency table and 'add' all the tables to fill contingency table 4.3. Then, the filled contingency table 4.3 after observing the corpus of figure 4.2 assuming incomplete data is displayed in table 4.7. Again, the empty cells in the table have the value $n_{ij} = 0$.

	ik	*jij*	*hij*	*at*	*wacht*	*kan*	*kunt*	
I	1.5			0.5	0.5	0.5		*3*
you		1.5		0.5	0.5		0.5	*3*
he			1.5	0.5	0.5	0.5		*3*
ate	0.5	0.5	0.5	1.5				*3*
wait	0.5	0.5			1			*2*
waits			0.5		0.5			*1*
can	0.5	0.5	0.5			1	0.5	*3*
	3	*3*	*3*	*3*	*3*	*2*	*1*	*18*

Table 4.8, The contingency table after 9 incomplete observations

The marginal totals are the same as the ones in table 4.5. Again, It is possible to estimate the probabilities p_{ij} of the translations from this table. This estimate will not be the maximum likelihood estimate, because we have distributed the observations of the phrases equally among the possible translations.

Actually, if we are dealing with incomplete data, no analytical solution is known for determining the maximum likelihood estimate. We can however use the estimation of p_{ij} from table 4.8 to fill a second contingency table. If we now are observing (*you can, jij kunt*) we can distribute this observation among the n_{ij} cells, using the estimations of p_{ij} from table 4.8. If we are taking several of these steps, the state of the contingency table will converge to a state in which it does not change significantly anymore. This is called the EM-algorithm and is defined in chapter 6.

Chapter 5

The translation model

In this chapter will examine the second step of the three basic steps mentioned in paragraph 3.2 more thoroughly. We have defined equivalence classes in the previous chapter, but we have not yet determined how to model each observation, and therefore we have not yet determined the definition of the probability distribution $P(E,D)$ with E an English sentence and D a Dutch sentence.

5.1 Introduction

Probabilistic models provide a theoretical abstraction of language. They are designed to capture the more important aspects of language and ignore the less important ones. The term model refers to some theory or conceptual framework about the observations that are made of the languages. Each model will contain parameters that represent the effects that particular variables or combinations of variables have in determining the values taken by the observations.

5.2 Information Theoretic approach

Shannon's theory of communication [Shannon, 1948], also known as *Information Theory* was originally developed at AT&T Bell Laboratories to model communication along a noisy channel such as a telephone line. Shannon was interested in the problem of maximising the amount of information that can be transmitted over a telephone line. The noisy channel paradigm can be applied to many linguistic problems, like for instance spelling correction, part-of-speech tagging of words and speech recognition.

It requires a bit more squeezing and twisting to fit the translation of phrases into the noisy channel architecture. To translate for example from Dutch to English, one imagines the noisy channel to be a translator who has thought up what he or she wants to say in English and then translates into Dutch before actually saying it.
Imagine we have an English sentence E. For example E = (*I*, *wait*). Suppose E is presented at the input to the noisy channel and for some crazy reason, it appears at the output of the channel as a Dutch sentence D. Our job is to determine E given D.

$$E \rightarrow \text{Noisy Channel} \rightarrow D \tag{1}$$

The most probable English sentence $\hat{E}$ given the Dutch sentence D is given by

$$\hat{E} = \underset{E}{\mathbf{argmax}}\, P(E)P(D|E) \tag{2}$$

The probability measure $P(E)$ is the prior probability. It is the probability that the translator will translate the English sentence E. $P(E)$ is a measure of what is likely to happen, and what is not

likely to happen. In a novel the sentence E = (*I, love, you*) is likely to happen. In a Master's thesis, however, we should hot expect to find that sentence. We can look at $P(E)$ as a 'grammar' that gives a relatively high probability to English sentences that are well formed and a relatively low probability to sentences that are ill formed. The distribution $P(D|E)$ can be looked upon as a English to Dutch dictionary (one that has information about all possible sentences D and E) that gives a relatively high probability to sentences that are each others translation.

Both probability distributions $P(E)$ and $P(D|E)$ are unknown and enormously complex. In this chapter we will define a model of the probability distributions $P(E)$ and $P(D|E)$ which we will call our *translation model*.

5.3 The prior probability: modelling sentences

Suppose that an English sentence E of length l is defined by a list of l English words so $E = (E_1, E_2,..., E_l)$. Then we can replace $P(E)$ by an model in which the probability of a sentence depends only on the probabilities of the separate words. This model is an abstraction or approximation of reality. However, within the model probabilities are exactly defined (and no approximations).

$$P(E) = P(E_1)\, P(E_2) \ldots P(E_l) \tag{3}$$

Because we do not use any sequence information or position information we can also use the well known multinomial distribution. Let the variable r be the number of equivalence classes, which we had defined in the previous chapter as the number of different English words. Using these equivalence classes we can define a random variables vector $N = (N_1, N_2,..., N_r)$ of the sentence $(E_1, E_2,..., E_l)$ in which N_i is the number of times the ith word of the sample space Ω_E appears in E. A sentence defined by N abstracts away from the position of the words and the sequence of the words. We look at a sentence as a collection of words, the order is not important. Suppose for example E = (*I, wait*) and Ω_E = {*I, you, he, ate, wait, waits, can*} then N = (1, 0, 0, 0, 1, 0, 0). The multinomial approximation of the prior probability $P(N)$ is given by

$$P(N_1 = n_1, \ldots, N_r = n_r) = \frac{l!}{n_1!\, n_2! \ldots n_r!}\, p_1^{n_1} p_2^{n_2} \ldots p_r^{n_r}, \quad n_i = 0,1,\ldots,l, \quad \sum_{i=1}^{r} n_i = l \tag{4}$$

The variables $p_1 \ldots p_r$ are the unknown parameters of the distribution. Actually the model presented in this paragraph is a very simple model. It is a first order model, so we do not look at context. More complex models can be used like for instance a bigram model which was used at IBM [Brown, 1993].

5.4 The channel probability: modelling translations

Suppose that the English sentence E is defined as a list of l English words and the Dutch sentence D is defined by a list of l Dutch words. So $E = (E_1, E_2,..., E_l)$ and $D = (D_1, D_2,..., D_l)$. Then we can replace $P(D|E)$ by a approximation in which each translated Dutch word depends only on one English word.

$$P(D|E) = P(D_1|E_1)\, P(D_2|E_2) \ldots P(D_l|E_l) \tag{5}$$

5.5 Drawing a parallel corpus

We have defined a translation model by defining approximations of the unknown probability distributions $P(E)$ and $P(D|E)$. By defining the approximations we have made various assumptions about how the parallel corpus 'was created' or how the parallel corpus 'came about'. Suppose the corpus was created like a lottery drawing and suppose the drawing takes place as defined by equations (3) and (5). We could describe our model by drawing different little balls from different urns. Each ball has his own probability that it is drawn from the urn. Drawing one parallel sentence takes five different steps:

1. The lottery mister or miss draws a ball from the urn containing 'sentence-length balls' determining which length l the sentences will be. The ball is drawn with probability $P(l)$.
2. The lottery mister or miss draws a ball from the urn containing 'English-word balls' determining what the first English word will be. The ball is drawn with probability $P(E_1)$.
3. The lottery mister or miss draws a ball from the urn containing 'Dutch-word-if-the-English-word-was-E_1 balls' determining what the first Dutch word will be. The ball is drawn with probability $P(D_1|E_1)$. There is a different urn for each English word E_1.
4. Step 2 and 3 are repeated for each next pair of English-Dutch words, l times in total.
5. Finally the words of the Dutch sentence are shuffled, so the sequence of English words may differ from the sequence of its Dutch translations.

Combining equations (3) and (5) we can define the joint probability distribution $P_{E\to D}(E,D)$ which is defined as the probability of drawing one parallel sentence if we are translating from English to Dutch.

$$P_{E\to D}(E,D) = P(l)\prod_{i=1}^{l} P(E_i)P(D_i|E_i) \tag{6}$$

5.6 Symmetry of the model

If we are able to find the most probable English translation E knowing the Dutch sentence D, we could of course make a similar model the other way around. However, the model we defined in this chapter has a special property. Because both $P(E)$ and $P(D|E)$ are approximated by a first order model (that is, they both depend only on the probability of the separate words), they are related by $P(E,D) = P(E)P(D|E)$. Therefore we can rewrite equation (6) as

$$P_{E\to D}(E,D) = P_{D\to E}(E,D) = \prod_{i=1}^{l} P(E_i,D_i) \tag{7}$$

In other words, it makes no difference if we are translating from English to Dutch or from Dutch to English. Either way we end up with the same translation model.

Again, we can define a 'multinomial like' model. Let the variable c be the number of different Dutch words. Let $M = (M_{ij} \mid 1 \le i \le r, 1 \le j \le c)$ be a stochastic matrix defined so that M_{ij} is the number of times the ith word of Ω_E is translated to the jth words of Ω_D in the pair of sentences (E,D) Suppose for example E = (*I, wait*) and Ω_E = {*I, you, he, ate, wait, waits, can*} and D = (*ik, wacht*) and Ω_D = {*ik, jij, hij, at, wacht, kan, kunt*} then

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{8}$$

$$P(M_{11}= m_{11}\dots M_{rc}= m_{rc}) = \frac{l!}{m_{11}!\dots m_{rc}!} p_{11}^{m_{11}}\dots p_{rc}^{m_{rc}}, \quad m_{ij} = 0\dots l, \qquad l = \sum_{i=1}^{r}\sum_{j=1}^{c} m_{ij}$$

The variables $p_{11}...p_{rc}$ are the unknown parameters of the distribution. The matrix M is the contingency table after observing the sentence pair (E,D) with complete data (see paragraph 4.4.1, table 4.4). In chapter 4 we saw that, depending on our knowledge of the language (that is our knowledge about the parameters $p_{11}...p_{rc}$) we can build a number of contingency tables. Not every table is as likely as the other. The probability that a contingency table M reflects what actually happened is given by equation (8).

Again, we can create a parallel corpus like a lottery drawing defined by equations (7) or (8). We could describe our model by drawing different little balls from only two different urns. Each ball has his own probability that it is drawn from the urn. Drawing one parallel sentence takes five different steps:

1. The lottery mister or miss draws a ball from the urn containing 'sentence-length balls' determining which length l the sentences will be. The ball is drawn with probability $P(l)$.
2. The lottery mister or miss draws a ball from the second urn containing 'English-Dutch-translation balls' determining what the first English and Dutch word will be. The ball is drawn with probability $P(E_1, D_1)$.
4. Step 2 is repeated for each next pair of English-Dutch words, l times in total.
5. Finally the words of the Dutch sentence are shuffled, so the sequence of English words may differ from the sequence of its Dutch translations.

The probabilities that a certain parallel corpus comes up are the same for this procedure as for the procedure the previous paragraph.

5.7 Discussion

In this chapter we have modelled the enormously complex things that happen in the mind of an author of an English text who translates his own text to Dutch. We have reduced it to a lottery drawing, or -maybe worse- to a noisy channel. A shortcoming of the noisy channel architecture is that it requires the statistical models to deal directly with English and Dutch sentences. Clearly the probability distributions are immensely complicated. On the other hand, in practice the statistical models must be relatively simple in order that their parameters can be reliably estimated from a manageable amount of training data. However, research at IBM has shown that surprisingly high translation accuracy can be achieved in practice using if the models $P(E)$ and $P(D|E)$ are restricted to the modelling of local linguistic phenomena.

The models $P(E)$ and $P(D|E)$ used at IBM are a bit more complex than the ones presented in this chapter. However the model we defined suffices for the objective of creating the bilingual dictionary. If we try to estimate the joint probability $P(E,D)$, we are able to calculate channel probabilities for both English to Dutch and Dutch to English dictionaries. If we try to estimate $P(D|E)$ via $P(E,D)$, we use more statistical information than Brown et al. did. We might therefore end up with a better estimation of $P(D|E)$ and $P(E|D)$.

Chapter 6

MLE from incomplete data: The EM-algorithm

In this chapter we will take the last step to determine the unknown probability measure $P(E,D)$: defining a good statistical estimator and estimation procedure. We will use Maximum Likelihood Estimation (MLE). Subsequently we will introduce an algorithm for computing the MLE from incomplete data. The algorithm is called Expectation Maximisation (EM-) algorithm. Its correctness was proven in 1977 by Dempster, Laird and Rubin. This chapter is based mainly on their article [Dempster, 1977].

6.1 Formal approach to MLE

The likelihood function L of S random variables $X^{(1)}$, $X^{(2)}$,...,$X^{(S)}$ is the joint probability distribution $P(X^{(1)}, X^{(2)},...,X^{(S)})$, which is the probability of an experiment of taking S random samples from the probability distribution $P(X)$. The probability distribution $P(X^{(1)}, X^{(2)},...,X^{(S)})$ is known except for k unknown parameters $\Phi = \{p_1, p_2,..., p_k\}$. To estimate the values of the unknown parameters we have to carry out the experiment and find realisations $x^{(1)}, x^{(2)},...,x^{(S)}$ of the random variables $X^{(1)},X^{(2)},...,X^{(S)}$. If we fill in the realisations in the likelihood function L then L may be considered as a function of the unknown parameters $\Phi = \{p_1, p_2,..., p_k\}$. If the likelihood function satisfies regularity conditions, the maximum-likelihood estimator is the solution of equation (1)

$$\frac{\mathrm{d}L(\Phi)}{\mathrm{d}\Phi} = 0 \tag{1}$$

In chapter 5 we defined equations (9) to be an approximation of the unknown translation probabilities $P(E,D)$. Remember that this approximation was given by

$$P(N_{11} = n_{11} \ldots N_{rc} = n_{rc}) = \frac{l!}{n_{11}! \ldots n_{rc}!} \prod_{i=1}^{r} \prod_{j=1}^{c} p_{ij}^{n_{ij}} \quad , n_{ij} = 0 \ldots l \quad , l = \sum_{i=1}^{r} \sum_{j=1}^{c} n_{ij} \tag{2}$$

in which the matrix N is a random variable that consists of the frequency counts n_{ij} of the (observed) translations and l is the length of the both the English and the Dutch sentence. Let the random variable X be the matrix N. Suppose we carry out the experiment described above by taken S random samples from the probability distribution of equation (2). Then the likelihood function L is given by

$$L = \prod_{s=1}^{S} \frac{l^{(s)}!}{n_{11}^{(s)}! \ldots n_{rc}^{(s)}!} \prod_{i=1}^{r} \prod_{j=1}^{c} p_{ij}^{n_{ij}^{(s)}} \quad , n_{ij}^{(s)} = 0 \ldots l^{(s)} \quad , l^{(s)} = \sum_{i=1}^{r} \sum_{j=1}^{c} n_{ij}^{(s)} \tag{3}$$

and its maximum occurs when

$$\hat{p}_{ij} = \frac{n_{ij}}{\sum_{s=1}^{S} l^{(s)}} \tag{4}$$

6.2 Criticism on MLE

The MLE is not always a suitable estimator. The problem is the sparseness of our data. While a few words are common, the vast majority of words are very uncommon. Valid translations of these words may very well be present in the corpus, but not all of them actually as a translation pair. The probability of a long English- Dutch sentence pair $P(E,D)$ is computed by multiplying the probabilities $P(E_i,D_i)$ that the words are translations. Because the MLE assigns zero probability to unseen events, one word pair in the test sentence, that was previously unseen in the training data, will give us bad (zero probability) estimates for the probability of valid sentences.

The sparseness of linguistic data from corpora may cause problems if we are estimating translation probabilities $P(E,D)$. Recently research is done to statistical estimators like Good-Turing estimation (see Appendix C for details) that assign some probability to unseen events. Still some questions remain.

1. will an information retrieval system benefit from the fact that unseen events may happen with some probability?
2. if so, how can the system choose the proper translation if it has a very low probability (because it is unseen)?

If a translation is unseen, the system will perform bad with particular queries, as the proper translation cannot be found if it has zero probability. So the answer to question one is yes.

If the approximation of the channel probability $P(D|E)$ allows unseen events to happen with very low probability then the approximation of the prior probability $P(E)$ has to make sure that, if necessary, the unseen translation is chosen. Consider for example a native speaker of Dutch wants to know something about *statistische automatische vertaling* (that is, statistical machine translation) and the approximation of the channel probability gives high probabilities to (*statistische* | *statistical*), (*automatische* | *automatic*) and (*vertaling* | *translation*) and a very low probability to (*automatische* | *machine*) because it was unseen in the training data. Of course the English word *automatic* is not the right translation in this context. If the approximation of the prior probability $P(E)$ is a bigram approximation, then it will probably assign very low probability to both (*statistical*, *automatic*) and (*automatic*, *translation*). A bigram approximation will probably assign relatively high probability to both (*statistical*, *machine*) and (*machine*, *translation*), choosing *statistical machine translation* to be the proper translation.

The sparseness of linguistic data from corpora may cause problems if we are estimating the channel probabilities $P(D|E)$. Estimators that assign some probability to unseen events may improve the translation system. However, we have to use a *n*-gram approximation of the prior probability $P(E)$ if we want to benefit from these estimators.

In this paper we will use simple MLE because

1. research in this paper is concentrated on the construction of the dictionaries $P(D|E)$ and $P(D|E)$ and not on the construction of language models $P(E)$ and $P(D)$,
2. the EM-algorithm is defined for MLE only and it is unknown how the algorithm behaves if other estimators are used.

6.3 Complete data vs. incomplete data

In the previous chapter we assumed that we could observe pairs of words which are each others translation. However, we cannot directly know by observation of pairs of sentences which words are each others translation. We assume that it is only known which *sentences* are each others translation. We refer to the observation of these sentences as *incomplete* observations. The incomplete observation problem can be visualised with a contingency table of which the marginal totals are known, but not the frequency counts of the cells itself (see chapter 4).

6.3.1 Definition of the incomplete data

The term incomplete data implies the existence of two sample spaces Ξ and Ψ and a many-to-one mapping from Ξ to Ψ. The observed data Y are a realisation from Ψ. The corresponding X in Ξ is not observed directly, but only indirectly through Y. We refer to X as the complete data. In our model the observed data or incomplete data $Y = y$ is defined by

$$Y = \begin{matrix} \{n_{i\cdot}^{(s)} | i = 1,2,\ldots r, s = 1,2,\ldots,S\} \\ \{n_{\cdot j}^{(s)} | j = 1,2,\ldots c, s = 1,2,\ldots,S\} \end{matrix}, \quad \sum_{i=1}^{r} n_{i\cdot}^{(s)} = \sum_{j=1}^{c} n_{\cdot j}^{(s)} = l^{(s)} \tag{5}$$

The observed data Y consists of the frequency counts $n_{i\cdot}^{(s)}$ of English words present in a sentence s together with the frequency counts $n_{\cdot j}^{(s)}$ of Dutch word present in the translation of s. The number of different English words is defined by r; the number of different Dutch words is defined as c. The length of both the English and the Dutch sentence is defined by $l^{(s)}$. The total number of sentences is denoted by S. Note that the English and Dutch vocabulary consist of respectively r and c words with r and c over thousand or ten thousand words. The mean sentence length $l^{(s)}$ is about 20 in the Agenda 21 corpus, so almost al of the counts $n_{i\cdot}^{(s)}$ and $n_{\cdot j}^{(s)}$ must be zero.

6.3.2 Definition of the complete data

The complete data $X = x$ in our model are the frequency counts of the (unknown) translations in each sentence.

$$X = \{n_{ij}^{(s)} | i = 1,2,\ldots,r; j = 1,2,\ldots c; s = 1,2,\ldots S\} \tag{6}$$

The complete data X is observed indirectly through the observed data Y by a mapping $Y \rightarrow Y(X)$ In our model $Y(X)$ is defined by.

$$Y(X) = \begin{cases} n_{i\cdot}^{(s)} = \sum_j n_{ij}^{(s)} \\ n_{\cdot j}^{(s)} = \sum_i n_{ij}^{(s)} \end{cases} \tag{7}$$

So given the counts of the unknown translations of words, we will also know which words were present in the sentences. Of course, given which words are present in a sentence s we do not know the translations. If we would the data would not be incomplete.

Both the complete data probability distribution $P(X)$ and the observed data probability distribution $P(Y)$ depend on the unknown parameters Φ of our model.

$$\Phi = p_{ij}, \quad i = 1,...,r; j = 1,...,c \tag{8}$$

The unknown probabilities Φ can be looked upon as the probabilistic dictionary we would like to have. The total number of different words is defined by r for English and by c for Dutch.

6.4 Definition of the EM algorithm

The EM algorithm is directed at finding a value of Φ which maximises the observed data probability distribution $P(Y)$, by making use of the complete data probability distribution $P(X)$. Given the incomplete data specification $P(Y)$ there are many possible complete data specifications $P(X)$ that will generate $P(Y)$. Each iteration of the EM algorithm involves two steps which we call the expectation step (E-step) and the maximisation step (M-step). Suppose that $\Phi^{(p)}$ denotes the current value of Φ and $T^{(p)}$ denotes the current value of $T(X)$ after p iterations of the algorithm. The function $T(X)$ denotes the complete data sufficient statistics (i.e. a possible smaller representation of X so that no information of X is lost).

E-step: Estimate the complete data sufficient statistics $t(X)$ by finding

$$T^{(p)} = E(T(X) \mid Y, \Phi^{(p)}) \tag{9}$$

M-step: Determine $\Phi^{(p+1)}$ as the solution of equation (10).

$$E(T(X) \mid \Phi) = T^{(p)} \tag{10}$$

6.5 Implementation of the E-step

Following Dempster's definitions of the E-step and the M-step we can apply the EM algorithm on our translation problem. First we will define the sufficient statistics $T(X)$ of the complete data X. After that we have to tackle the alignment problem to implement the E-step. Because simple counting down all possible combinations probably will take too much processing time, we have to look into different solutions.

6.5.1 Definition of the sufficient statistics

It is not necessary to keep track of every expected value of $n_{ij}^{(s)}$. The total frequency counts n_{ij} after S observations are sufficient to estimate the parameters Φ of our model. Therefore we define the complete data sufficient statistics to be

$$T(X) = \{n_{ij} \mid n_{ij} = \sum_{s=1}^{S} n_{ij}^{(s)}; i = 1 \ldots r; j = 1 \ldots c\} \tag{11}$$

The data sufficient statistics $T(X)$ can be displayed in a contingency table (see paragraph 6.2). Now we can apply the E-step after p iterations as follows:

$$T_{ij}^{(p)} = E(n_{ij} \mid n_{1\cdot}^{(1)} \ldots n_{r\cdot}^{(1)}, n_{\cdot 1}^{(1)} \ldots n_{\cdot c}^{(1)}, \ \ldots \ , n_{1\cdot}^{(s)} \ldots n_{r\cdot}^{(s)}, n_{\cdot 1}^{(s)} \ldots n_{\cdot c}^{(s)}, \Phi^{(p)}) \tag{12}$$

Because the expectation of a sum is equal to the sum of the expectation, and because the marginal totals of one sentence does not influence the translation of others, we can calculate the expectation for each sentence separately by

$$T_{ij}^{(p)} = \sum_{s=1}^{S} E(n_{ij}^{(s)} \mid n_{1\cdot}^{(s)} \ldots n_{r\cdot}^{(s)}, n_{\cdot 1}^{(s)} \ldots n_{\cdot c}^{(s)}, \Phi^{(p)}) \tag{13}$$

The sufficient statistic we defined is sufficient to calculate the probabilities from the frequency counts (which is, as we will see in paragraph 6.6, the M-step). However, $T(X)$ is not sufficient to calculate the E-step. We can only calculate the E-step if we look separately at every sentence in the corpus.

6.5.2 Counting down all combinations

The average number of words in an Agenda 21 sentence is about 20. The number of possible separate events of equation (2) (the number of possible alignments) increases exponentially with the length l of both sentences. If both source and target language sentences consist of only 10 words, then the number of possible alignments with independent source and target language words is 10! = 3,628,800. It seems, therefore that it is not feasible to evaluate the expectation in equation (13) exactly.

The number of possible alignments with independent source language words, which was used at IBM [Brown, 1990], is 10^{10} = 10,000,000,000. In this case it is possible to use a 'trick' called a *combinatorial generating function* [Mood, 1963]. The subject of combinatorial generating functions is a field of mathematics itself, and we shall consider only the generating function used at IBM as an example. Suppose we want to align two sentences of length l = 3. Then the number of possible alignments with independent source language words is 3^3 = 27. All possible alignments are given by the sum

$$p_{11}p_{21}p_{31} + p_{11}p_{21}p_{32} + ... + p_{13}p_{23}p_{32} + p_{13}p_{23}p_{33} \tag{14}$$

The 27 terms in equation (14) are the expansion of the generating function given by

$$(p_{11}+p_{12}+p_{13})\,(p_{21}+p_{22}+p_{23})\,(p_{31}+p_{32}+p_{33}) \tag{15}$$

To calculate equation (14) we need 81 multiplications and additions, but calculating equation (15) only needs 8 multiplications and additions. Because the alignment with independent source language words is not symmetric, we will not use the algorithm used at IBM.

6.5.3 Combining equivalence classes

We do not know if a generating function exist for the combinations of equation (2). In a strange way the generating function of equation (15) seems to combine all the equivalence classes of the target language to one single class. To analyse contingency tables, the combination of equivalence classes may sometimes help to make better statistical inference [Everitt, 1992].

Suppose we want to calculate the expected translation n_{ij} of (*love*, *liefde*) from the sentence pair ("*love and peace*", "*liefde en vrede*") with equation (13). The E-step can be calculated simple if we were to combine the event *and* with *peace* and the event *en* with *vrede*, reducing the possible events to *love*, NOT(*love*), *liefde* and NOT(*liefde*).

	liefde	*en*	*vrede*	
love	n_{ij}	**?**	**?**	*1*
and	**?**	**?**	**?**	*1*
peace	**?**	**?**	**?**	*1*
	1	*1*	*1*	*3*

$\rightarrow$

	liefde	NOT(*liefde*)	
love	n_{ij}	1- n_{ij}	*1*
NOT(*love*)	1- n_{ij}	1+ n_{ij}	*2*
	1	*2*	*3*

Table 6.1, combining equivalence classes

Because we combined the equivalence classes, the contents of the contingency table are known except for the expected translation n_{ij} of (*love*, *liefde*). In general it can be proven that for each sentence the expected counts for the random parameters $N_{ij} = n_{ij}$ are given by

$$E(N_{ij}|n_{1\cdot}, n_{\cdot 1}, \ldots) = \sum_{x=\max(0,n_{i\cdot}+n_{\cdot j}-n)}^{\min(n_{i\cdot},n_{\cdot j})} C \cdot x \binom{l-n_{i\cdot}}{n_{\cdot j}-x}\binom{n_{i\cdot}}{x}\left(\frac{p_{ij}(p_{\cdot\cdot}-p_{i\cdot}-p_{\cdot j}+p_{ij})}{(p_{i\cdot}-p_{ij})(p_{\cdot j}-p_{ij})}\right)^x \tag{16}$$

The constant C is determined by the constraint $\sum P(N_{ij}) = 1$ and can therefore be computed numerically.

We have implemented the EM-algorithm defined by equation (16). Tests show that this approximation of the E-step gives bad estimates of the translation parameters. The bad estimates result from the combination of equivalence classes of which the probabilities differ considerably from each other. Function words like *and* and *en* in the example given above occur possibly a thousand times for every occurrence of *peace* or *vrede*. Because function words like *and* literally occur in every sentence, equation (16) is of no practical use. You can find more about the performance in paragraph 6.7.

6.5.4 Iterative proportional fitting

A very old way to find missing values in $n \times n$ contingency table is the *iterative proportional fitting procedure* (IPFP). This algorithm appears to have been first described in 1937 by Kruithof. The basic IPFP takes an contingency table with initial counts $n_{ij}^{(0)}$ and sequentially scales the table to satisfy the observed data $m_{i\cdot}$ and $m_{\cdot j}$. We assume that the marginal totals $n_{i\cdot}^{(0)}$ and $n_{\cdot j}^{(0)}$ are not yet in correspondence with the observed data $m_{i\cdot}$ and $m_{\cdot j}$. The pth iteration of the consists of two steps which form:

$$n_{ij}^{(p,\,1)} = n_{ij}^{(p\text{-}1,\,2)} \cdot m_{i\cdot} / n_{i\cdot}^{(p\text{-}1,\,2)}$$
$$n_{ij}^{(p,\,2)} = n_{ij}^{(p,\,1)} \cdot m_{\cdot j} / n_{\cdot j}^{(p,\,1)} \qquad (17)$$

The first superscript refers to the iteration number, and the second to the step number within iterations. The algorithm continues until the observed data $m_{i\cdot}$ and $m_{\cdot j}$ and the marginal totals $n_{i\cdot}^{(p)}$ and $n_{\cdot j}^{(p)}$ are sufficiently close.

Is it possible to forget all about the EM-algorithm and estimate the parameters with the IPFP?. No, the IPFP is only applicable if we have a realistic initial guess $n_{ij}^{(0)}$ of the frequency counts. If we do not have any knowledge of the languages we model, the best guess we can possibly make, is dividing the frequency counts equally among the possible translations (see table 6.7). If we however divide the frequency counts equally among the possible translations, the observed data $m_{i\cdot}$ and $m_{\cdot j}$ already fits the marginal totals. The IPFP will be converged before it has started.

Because the IPFP provides an estimate that of the frequency counts that fulfils the observed marginal totals, we can use the IPFP to replace the E-step of the EM-algorithm. This leaves us with one question: What do we take as an initial guess $n_{ij}^{(0)}$ of the frequency counts of the IPFP. The most obvious choice seems to take the initial guess direct from the current estimate p_{ij} of the joint probabilities.

$$n_{ij}^{(0)} = p_{ij} \frac{l}{\sum_{i,j}^{l} p_{ij}} \qquad (18)$$

The variable l is the length of the sentence we are dealing with. If we are taking equation (18) as an initial guess, we can use the IPFP to fit the complete data to the observed, incomplete data. Taking equation (18) as an initial guess will give the same results as taking $n_{ij}^{(0)} = p_{ij}$ as an initial guess.

6.5.5 Brown's E-step

So far, we have mentioned research at IBM [Brown, 1990, 1993] so often in this paper, that we may want to look some more at their statistical MT attempt. How did Peter Brown et al. implement their E-step?

Brown et al. used a different approach in three ways. Remember first that they tried to estimate the conditional probability measure $P(F|E)$ of an English sentence E and a French sentence F directly, as they were not interested in a bi-directional translation model (We try to estimate the joint probability measure $P(E,D)$). Remember also that they used an alignment model with independent English words, allowing multiple French words to correspond with one English word. Remember finally that they used a generating function to resolve from the combinatorial explosion of possible connections. As a result they found the following E-step of their EM-algorithm.

$$E(N_{ij}|n_{i.},n_{.j}) = \frac{t(f_i|e_j)}{t(f_i|e_1)+t(f_i|e_2)+\ldots+t(f_i|e_l)} n_{i.}\, n_{.j} \tag{19}$$

We adopted the conditional probability parameter notation $t(f_i|e_j)$ from Brown et al. to indicate the difference with our joint probability parameter p_{ij}.

Although Brown et al. used a different approach in three ways and although they did not once mention the IPFP, the resemblance of both E-steps is striking. In fact, their E-step can be seen as one IPFP iteration, leaving some errors in the marginal totals of the French words (which is not a problem because of their unidirectional approach).

We can look at the conditional parameter $t(f_i|e_j)$ as the first step of one IPFP iteration, because calculating the conditional probability from the joint probability requires division by the marginal probability, just like the first step of one IPFP iteration, if the joint probabilities are taken as an initial guess. The difference is that the overall conditional probability is taken and not the per-sentence probability. The division in equation (18) is exactly the second step of one IPFP iteration.

6.5.6 A dirty trick

We wanted to know if equation (18) was a good initial guess of the frequency counts and experimented with some other initial estimates. Inspired by equation (16) we defined with the assumption that we could take equivalence classes together, we used the following as an initial estimate of the frequency counts of the IPFP.

$$n_{ij}^{(0)} = \frac{p_{ij}(p_{..} - p_{i.} - p_{.j} + p_{ij})}{(p_{i.} - p_{ij})(p_{.j} - p_{ij})} \tag{20}$$

With this initial guess, we performed the IPFP as described in paragraph 6.5.4 to replace the E-step of the EM-algorithm.

6.6 Implementation of the M-step

Equation (10) is the familiar form of the likelihood equations for maximum-likelihood estimation That is, if we were to suppose that $T^{(p)}$ represents the sufficient statistics computed from an observed X drawn from an regular exponential family, then equation (4) defines the maximum-likelihood estimator of Φ. The M-step is therefore given by equation (4).

6.7 Comparing the different algorithms

During the project we were not able to make an objective comparison between the different E-steps we formulated in this chapter. However, we can give an impression of the overall performance by giving some example estimations of the algorithm. In this chapter we will give

the most probable Dutch translations of some English words that appear a large number of times in the corpus as an indication of the performance of the algorithm. We will show the estimates of two function words *and* and *the*. Additionally we show the estimates of two ordinary words: *health* and *training*. The reader must hold in mind that these estimates only give an indication of the performance of the algorithms.

6.7.1 Combining equivalence classes

This is the algorithm presented in paragraph 6.5.3. The estimates are particularly bad on the non-function words. On function words like *and* and *the* the algorithm seems to do fine. However, the probability estimates of the translations of *health* and *training* are much to low.

and		*the*		*health*		*training*	
en	*0.95*	*de*	*0.63*	*gezondheid*	*0.08*	*scholing*	*0.08*
als	*0.01*	*het*	*0.13*	*gezondheidszorg*	*0.06*	*op*	*0.04*
(null)	*0.01*	*(null)*	*0.05*	*milieu*	*0.04*	*opleidingen*	*0.03*
		in	*0.03*	*volksgezondheid*	*0.03*	*(null)*	*0.03*
		op	*0.02*	*het*	*0.03*	*opleiding*	*0.03*
		een	*0.01*	*in*	*0.02*	*scholingsprogramma*	*0.02*

figure 6.2, example entries

The six most probable translations of the English words are given in table 6.2. In the first column the possible Dutch translation is given, *(null)* means that the English word is not translated to Dutch. In the second column the probability of the translation is given. If the probability was less than 0.005, than the possible translation is not displayed. The remaining 3% of the probable Dutch translations of *and* is divided among a number of unlikly translations.

6.7.2 IPFP with p_{ij} as initial estimate

This is the algorithm presented in paragraph 6.5.4. This algorithm seems to behave better on the non-function words. However, on the function words the algorithm behaves not that good. It even gives more probability to *van* (i.e. *of* in English) being the translation of *the* than to *de* being the translation of *the*.

and		*the*		*health*		*training*	
en	*0.80*	*de*	*0.35*	*gezondheid*	*0.27*	*scholing*	*0.26*
van	*0.05*	*van*	*0.16*	*gezondheidszorg*	*0.16*	*opleidingen*	*0.12*
de	*0.04*	*het*	*0.13*	*volksgezondheid*	*0.08*	*opleiding*	*0.10*
voor	*0.02*	*in*	*0.05*	*de*	*0.06*	*(null)*	*0.07*
het	*0.01*	*te*	*0.05*	*gezondheidsprobl.*	*0.04*	*scholingsprogramma*	*0.05*
te	*0.01*	*(null)*	*0.04*	*gezondheids*	*0.04*	*en*	*0.04*

figure 6.3, example entries

6.7.3 Brown's E-step

This is the algorithm presented in paragraph 6.5.5. This algorithm seems to behave very well on the non-function words. We showed only 6 possible translations of the English words, but of the first 10 possible Dutch translation of *health* 9 are Dutch compounds beginning or ending with '*gezondheid*' All of the first 10 possible translations of *health* are Dutch compounds beginning with '*opleiding*' or '*scholing*'.

On the function words, however, the algorithm behaves bad. It gives again more probability to *van* (i.e. *of* in English) being the translation of *the* than to *de* being the translation of *the*.

and		*the*		*health*		*training*	
en	*0.48*	*de*	*0.22*	*gezondheid*	*0.27*	*scholing*	*0.26*
van	*0.07*	*van*	*0.12*	*gezondheidszorg*	*0.16*	*opleidingen*	*0.12*
de	*0.04*	*het*	*0.08*	*volksgezondheid*	*0.08*	*opleiding*	*0.11*
(null)	*0.04*	*in*	*0.04*	*gezondheidsprobl.*	*0.06*	*scholingsprogramma*	*0.07*
het	*0.03*	*(null)*	*0.04*	*gezondheids*	*0.04*	*opleidings*	*0.05*
voor	*0.03*	*te*	*0.03*	*gezondheidsrisico*	*0.04*	*opleidingsmogelijkh.*	*0.04*

figure 6.4, example entries

6.7.4 IPFP with 'dirty trick' initial estimate

This is the algorithm presented in paragraph 6.5.6. The algorithm seems to behave as well on the function words as on the non-function words.

and		*the*		*health*		*training*	
en	*0.93*	*de*	*0.68*	*gezondheid*	*0.28*	*scholing*	*0.28*
zijn	*0.01*	*het*	*0.14*	*gezondheidszorg*	*0.20*	*opleidingen*	*0.12*
als	*0.01*	*(null)*	*0.03*	*volksgezondheid*	*0.11*	*opleiding*	*0.11*
(null)	*0.01*	*in*	*0.02*	*gezondheidsprobl.*	*0.05*	*scholingsprogramma*	*0.08*
		te	*0.01*	*gezondheids*	*0.04*	*(null)*	*0.07*
		aanmerk.	*0.01*	*te*	*0.02*	*opleidings*	*0.04*

figure 6.5, example entries

6.8 Discussion

Comparing the performance of different EM-algorithms just by looking at the probability estimates is not a very objective method. If statistical models have to be tested, we usually look if it is capable of 'explaining' sentences it has never seen before. We are aware it is hard (or even impossible) to predict from table 6.2, 6.3, 6.4 and 6.5, which estimates will perform the best. However, we believe that the dictionary entries we presented above give an indication of the usefulness in later applications. It seems we developed with our 'dirty trick' approach an algorithm that provides better estimates than Brown's EM-algorithm (which is not a symmetric algorithm and will produce only an accurate English-to-Dutch dictionary and not an accurate Dutch-to-English dictionary). We have however no theoretical proof that the IPFP may be used to replace the E-step. We have also no theoretical proof that equation (20) may be used as an initial guess of the E-step. Until we have this proof it seems reasonable to call the algorithm presented in 6.5.6 a 'dirty trick'.

In this paragraph we would again like to emphasise that there exists a principle difference between Brown's EM-algorithm and our EM-algorithm (see paragraph 6.5.5). Brown et al. tried to estimate the conditional probability measure $P(E|D)$, but we tried to estimate the joint probability measure $P(E,D)$, making it easy to switch from $P(E|D)$ to $P(D|E)$ if we are going to use the estimates as a dictionary. We believe that estimating $P(E,D)$ instead of $P(E|D)$ has the additional advantage that it is possible to give more accurate estimates. Like said before, we have no theoretical proof of this hunch. However, the performance of the EM-algorithm introduced in paragraph 6.5.6 indicates that better estimates are possible.

Chapter 7

Evaluation using Agenda 21

In this chapter we will look at the results of the evaluation of our EM-algorithm. We will use the English and the Dutch version of Agenda 21 as a parallel corpus. First we will give a description of the experiment we will take. After that, we will look at the results of the evaluation. Some technical details of the design steps can be found in appendix D.

7.1 The experiment

The experiment consists of 4 steps. First we divided the Agenda 21 in a training corpus and a testing database. Secondly, we trained the parameters of our model. After that, we asked volunteers to look for fragments of Agenda 21 in the testing database. Finally we perform recall and precision measures on the retrieved fragments.

7.1.1 Dividing the corpus

Evaluating statistical models involves a training step and a testing step. To make a valid statement of the model possible, two corpora have to be used: one corpus as a training corpus and another as a testing corpus. We cannot use the entire Agenda 21 corpus both as a training and as a testing corpus. Due to the fact that the model is already optimised on the test corpus, the outcome of testing is much better than it would be for any other corpus. Therefore we divided Agenda 21 in two parts, using only one part for training the parameters of our model.

7.1.2 Training the parameters

We trained the parameters of our model with the EM-algorithm defined in chapter 6. We used the variant of the E-step defined in paragraph 6.5.6. The EM-algorithm presented in chapter 6 expects the English and the Dutch sentences to be of equal length. This is not a very realistic assumption, as the average sentence length of the English Agenda 21 corpus is 20.8 words and the average sentence length of the Dutch corpus is 24.5 words. To be able to perform the EM-algorithm properly, we make the assumption that some words are not translated at all. To model this assumption we introduce for each language a special *(null)* word. If the length of, for example, the English sentence is smaller than the length of the Dutch sentence, the English sentence is filled up with the special *(null)* words.

7.1.3 Using the testing corpus as a document base

We wanted to know if our translation system might be useful for information retrieval purposes. Therefore we needed a multi-lingual database containing different documents of the same genre as the model is trained with. We can achieve this by using the testing corpus as a database containing different fragments of the Agenda 21 corpus. Note that of each fragment we have an English and a Dutch translation.

The retrieval system

We built a retrieval system based on a Boolean IR model. In a Boolean IR model the document is represented by a collection of *catch words*. A query consists of a list of catchwords separated by the Boolean operators like AND, OR and NOT. Query and documents can be matched by checking if the catch words that belong to the documents will make the query true [Hemels, 1994]. The system processes each Dutch query as follows. First the Dutch query is stripped from phrases like for example ("*I want to know more about...*"). After that, the system uses the non-function words from the query and the AND operator to produce a Boolean query. For example the natural language query "*I am interested in overpopulation problems in central Africa*" will be reduced by the system to the Boolean query: *overpopulation* AND *problems* AND *central* AND *Africa*.

The experiment

We used an experiment that is inspired by Mauldin's evaluation on a knowledge-based document retrieval system [Mauldin, 1991]. The experiment was conducted with a number of persons that operated the document retrieval engine. The native language of the volunteers is Dutch. The experiment was organised as follows.

1. First, each person is given some time to look at the Dutch Agenda 21 corpus to give them an idea of the topics in the database that may be retrieved. This is done to be sure that the person does not try to retrieve information that is not available in the database. Mauldin went a step further by showing each person the document he or she had to retrieve.
2. Each person is asked to formulate a Dutch query of their information need.
3. The Dutch query is used to retrieve documents from the Dutch database.
4. The Dutch query is translated to its most probable English translation using the probabilistic dictionary constructed with the training part of Agenda 21.
5. The English query is used to retrieve documents from the English database.
6. The retrieved Dutch documents, together with the Dutch translations of the retrieved English documents, are presented to the user. The user has to decide of each retrieved document if it is relevant or not.

The volunteers are only confronted with the Dutch version of Agenda 21, with Dutch queries and with Dutch documents that they retrieved. Therefore, their ability to speak, or translate from, English (or any human's ability to translate from English) does not effect the experiment.

7.1.4 Measuring retrieval performance

Effectivity of an information retrieval (IR) system is traditionally measured by *recall* and *precision*. Recall is the fraction of the relevant documents, that is actually retrieved. Precision is de fraction of the retrieved documents, that is actually relevant. It is often simple to obtain high recall at the cost of the precision. Likewise it is relatively simple to obtain a high precision at the cost of recall. Finding a good balans between recall and precision is the real problem of information retrieval.

Precision can be calculated fairly easily. Recall is harder to calculate. The entire database has to be examined to determine the recall exactly. If the performance of different IR systems has to be compared, *relative recall* may be used as a performance measure. The relative recall of system 1 with regard to system 2 is the fraction of the relevant documents retrieved by both systems, that is actually retrieved by system 1.

More recent IR performance measures depend more heavily on the opinion of the user and on the willingness of the user to browse through lists of possible relevant documents. This leads to

requirements for successful document retrieval called *prediction criterion* and *futility point criterion* [Hemels, 1994].

7.2 The results

Before we are actually going to analyse Agenda 21 we might want to look at some of the global characteristics of the corpus. After that we will look at the training algorithm and give some preliminary results of the dictionary we have created. Finally we will give the results of the retrieval experiment.

7.2.1 Global corpus characteristics

Just like Brown's experiment at IBM [Brown, 1993], we will use as a training corpus the sentences of Agenda 21 that have a maximum length of 30 words. The remaining sentences will be used to simulate the document base. We will call each different word in the corpus (actually every equivalence class) a new *type*. Each occurrence of a word will be called a *token* (see table 7.1).

Characteristics	total corpus	train corpus	train / total
tokens	146089	59419	40.7 %
types	5385	3854	71.6 %
sentences	7022	4664	66.4 %

table 7.1, size of the English Agenda 21 train and test corpus

Because we took only the smaller sentences, we used almost two third of the sentences of the entire corpus, but only 40.7% of the words of the entire corpus.

Characteristics	total corpus	train corpus	train / total
tokens	172186	68026	39.5 %
types	8518	5462	64.1 %
sentences	7022	4664	66.4 %

table 7.2, size of the Dutch Agenda 21 train and test corpus

Table 7.2 represents the characteristics of the Dutch Agenda 21 part. Of course the number of sentences of both corpora (after sentence alignment) are the same. Note that in the Dutch version, the number of types exceeds the number of sentences. In both the English and Dutch part a considerably part of the types does not appear in the train set at all (respectively 28.4% and 35.9%). This gives the impression that the Agenda 21 may be too small to make good statistical inference for the domain of ecology and sustainable development.

Table 7.3 gives us the size of some well known bilingual corpora, starting with the famous Rosetta stone, that Champollion used to decipher the ancient Egyptian hieroglyphs. The Hansard corpus consist of the Canadian parliamentary debates that are available in both English and French. The Hansards were used by many of the researchers mentioned in chapter 3. The Hansard corpus is more than 500 times bigger than the Agenda 21 corpus.

Corpus	Size (×1000 words)	nr. of translations
Rosetta stone	0.1	3
Agenda 21	170	± 80
Bible	1,000	> 150
Shakespeare	2,000	> 40
Hansard	90,000[1]	2

table 7.3, the size of some multi-lingual corpora

The Agenda 21 corpus is probably too small to make good statistical inference. However, we may still use it to indicate what would be possible if we used a bigger bilingual corpus.

7.2.2 Some preliminary results

We implemented the EM-algorithm as stated in paragraph 6.5.6. After 6 training steps of the algorithm the parameters do not change significantly anymore. Now let us take a quick glimpse at some of the results. First we will look at the possible translations of some of the English function words that appear early in the alphabet. These words appear at least 100 times (for *also*) till more than 4000 times (for *a*) in the corpus. Then we will look at the way the algorithm behaves if it is confronted with certain linguistic phenomena (see appendix B, for explanation of the linguistic terms).

a		*also*		*and*		*be*	
een	0.69	*ook*	0.61	*en*	0.93	*worden*	0.69
(null)	0.06	*tevens*	0.15	*zijn*	0.01	*zijn*	0.08
het	0.04	*eveneens*	0.13	*als*	0.01	*te*	0.04
aan	0.02	*daarnaast*	0.03	*(null)*	0.01	*(null)*	0.04
die	0.02	*bevatten*	0.01			*de*	0.02
te	0.02	*evenals*	0.01			*komen*	0.01

Figure 7.4, example entries of some English function words

The six most probable translations of the English words are given in table 7.4. In the first column the possible Dutch translation is given, *(null)* means that the English word is not translated to Dutch. In the second column the probability of the translation is given. If the probability was less than 0.005, than the possible translation are not displayed. The remaining 3% of possible translations of *and* in table 7.4c is divided over a number of possiblilities with low probability.

een		*ook*		*en*		*worden*	
a	0.41	*also*	0.53	*and*	0.96	*be*	0.60
(null)	0.22	*(null)*	0.21	*(null)*	0.01	*(null)*	0.12
an	0.12	*including*	0.05			*are*	0.10
the	0.03	*development*	0.03			*as*	0.02
of	0.02	*include*	0.02			*to*	0.01
one	0.02	*on*	0.02			*of*	0.01

Figure 7.5, example entries of some Dutch function words

[1]Klavans and Tzoukerman reported the size of the Hansards to be 85 million English and 95 French words [Klavans, 1990]. Today the corpus is even bigger. Brown et all accually used 29 million words so still about 170 times as much as the total size of our corpus [Brown, 1993].

All these function words are each others most probable translation. So, the most probable English translation of the Dutch word *een* is *a* and the most probable translation of the English word *a* is the Dutch word *een*. This is not necessarily the case as we will see in figure 7.7.

local	
plaatselijke	0.51
lokale	0.24
lokaal	0.15
plaatselijk	0.09
maken	0.01

can	
kunnen	0.58
kan	0.33
dit	0.03
leveren	0.03
brede	0.01

dieren	
animal	0.50
animals	0.40
(null)	0.08
such	0.01

verbetering	
improving	0.31
improvement	0.28
improve	0.16
improved	0.06
enhancing	0.03
(null)	0.02

Figure 7.6, example entries of morphlogically realted words and synonyms

unsustainable	
duurzame	0.57
niet	0.33
voorkomen	0.03
trekken	0.02
onhoudbaar	0.02
een	0.02

duurzame	
sustainable	0.93
unsustainable	0.02
renewable	0.02
consumption	0.01
sustainability	0.01

Figure 7.7, example entries of English morphology

These entries explain why the performance of the monolingual Dutch retrieval engine will probably differ considerably (as we will see) from the performance of the multilingual English-to-Dutch retrieval engine. Even if the words are translated correct, a lot of possibly correct translations are not used.

volksgezondheid	
health	1.00

health	
gezondheid	*0.28*
gezondheidszorg	*0.20*
volksgezondheid	*0.11*
gezondheidsprobl.	*0.05*
gezondheids	*0.04*
te	*0.02*

Figure 7.7, example entries of compounds

Unlike the English *unsustainable* (i.e. *niet* duurzame) which was has both *duurzame* and *niet* as probable translations. The Dutch word *volksgezondheid* (i.e. *people's* health) is has only *health* as a (certain) translation. The most probable translation of *health*, however is correctly *gezondheid*.

7.2.3 The multilingal IR results

In this paragraph we will give the results of the experiment described in paragraph 7.1. We were able to get the cooperation of 8 volunteers, all with Dutch as their native language. They formulated a total of 41 Dutch queries that were used to extract fragments from both the English and the Dutch Agenda 21 test corpus.
The test corpus consists of 2358 sentences of the Agenda 21 corpus. If two sentences followed each other in the corpus they were taken together as one fragment. This way we were able to construct a multilingual database of 1545 English fragments together with their Dutch translations.

Translation of the queries was done with the estimates described in the previous paragraph. If a word of the query did not have an entry in our dictionary we used that (Dutch) word to search the English database.

Some problems may occur if the systems retrieves a large number of fragments. We decided that the volunteers should not have to judge more than 15 retrieved fragments. If more than 15 fragments were retrieved, only the first 15 were showed to the volunteers. This may however give serious distortions in the relation between the number of retrieved English fragments and the number of retrieved Dutch fragments. For example, if the query retrieves 60 English fragments and 20 Dutch fragments, showing the first 15 fragments might lead to showing 15 English fragments and none of the Dutch. If this happened we would occasionally skip some of the English fragments and show the volunteers about 10 English fragments (that is the Dutch translations) and 5 Dutch fragments.

per-son	total	total OK	English	English OK	precision	relative recall	Dutch	Dutch OK	precision	relativ e recall
1	53	27	50	26	0.520	0.963	13	9	0.692	0.333
2	48	43	44	39	0.866	0.907	24	22	0.917	0.517
3	10	7	6	5	0.833	0.714	4	2	0.500	0.286
4	60	34	51	27	0.529	0.794	25	15	0.600	0.441
5	0	0	0	0	-	-	0	0	-	-
6	34	30	30	26	0.867	0.867	15	15	1.000	0.500
7	56	39	42	26	0.619	0.667	38	30	0.789	0.769
8	8	2	8	2	0.250	1.000	0	0	-	-
tot.	269	184	231	151	0.654	0.821	119	93	0.782	0.511

Table 7.8, Results of the IR experiment

The results per volunteer are given in table 7.8. Each row of table 7.8 gives the results of one volunteer. The last row gives the total results. The second and the third column give respectively the total number of fragments retrieved and the number of fragments that contained information the volunteer was looking for. The next four columns give the results on the English database. The last four columns give the results on the Dutch database.

Surprisingly, it seems that the system performs better on the English database than on the Dutch database, even if the translation are not always accurate (the system translates for example the query *chemische stoffen* to *chemical chemicals*). Indeed, the precision of the multilingual Dutch-to-English retrieval system is a bit lower than the precision of the monolingual Dutch retrieval system, respectively 65% and 78%. The relative recall of the multilingual system, however, is much higher than the relative recall of the monolingual retrieval system, respectively 82% and 51%. In the next paragraph we will try to find explanations for these results.

7.2.4 Discussion of the multilingual IR results

To find a good explanation of the results we first will look at the English translations that the system generated. We assigned each of the resulting English queries a category according to the following criteria. If the translated query was translated correct we assigned it to the *correct* category. If the query was translated incorrect, but was able to retrieve correct fragments the query we assigned it to the *usable* category. If the query was translated incorrect because only a part of the original was translated, we assigned it to the *partially correct* category. If the query could not be translated at all, because the words in the query were not present in the dictionary,

we assigned it to the *not translated* category. Finally, we assigned the remaining queries to the *incorrect* category.

Of the 41 queries 19 fell into the correct category, 3 fell into the usable category, 10 fell into the partially usable category, 6 fell into the not translated category and 6 fell into the incorrect category. We feel that a translated sentence that is in any of the first two categories (correct or usable) represents a reasonable translation. By this criterion the system performed successfully 54% of the time. Only 6 out of 41 is 15% of the queries were translated incorrect.

correct	
Dutch query	*afspraken over samenwerking tussen verschillende landen*
Translated as:	*arrangements on cooperation between different countries*
usable	
Dutch query	*gezondheid van de mens*
Translated as:	*health of the human*
partially usable	
Dutch query	*verbeteren van de milieubescherming*
Translated as:	*improve of the protection*
not translated	
Dutch query	*het kappen van regenwouden in de Filipijnen*
Translated as:	**?** *of* **?** *in the* **?**
incorrect	
Dutch query	*het aandeel van windenergie tot het totaal van energiebronnen*
Translated as:	*giving of* **?** *irrigated of energy*

Figure 7.9, Translation examples

Of course, incorrect translations will decrease both precision and recall of the multilingual retrieval system.
Usually queries that fell into the not translated category, did not retrieve any fragments from the Dutch database either. So the queries that fell into the not translated category did not influence the performance of both systems much.

Queries that fell into the partially usable category often contain Dutch compounds (see appendix B) that ought to be translated to two separate English words. Our translation model is only able to find one of these words, possibly increasing the recall, but decreasing the precision of the system. The reason that this often leads to an improvement of the performance is the limitation of our domain and the limitation of our corpus. For example, the query *armoedebestrijding* (i.e. *combating poverty*) is, because it is a Dutch compound, translated to *poverty*. However, if we are talking about poverty in the domain of Agenda 21, we usually talk about the *combating of poverty*. If our database contained fragments of other domains, the recall would not be increased as much as it did now.

Still, the phenomena above do not explain why the multilingual retrieval system seems to perform *better* than the monolingual Dutch retrieval system. It seems that there is a more structural reason. If we look again at table 7.1 and table 7.2 we see that the English corpus contains 5385 different words and the Dutch corpus 8518 different words. The difference can

be explained by Dutch compounds, but also by the use of more synonyms in Dutch, by the richer Dutch morphology and by the existence of translational ambiguities in the translation of English to Dutch (see appendix B).

Dutch synonyms

One of the volunteers formulated the query *plaatselijke Agenda 21*, which was translated by the system to *local Agenda 21*. In English *local Agenda 21* is the term used throughout the corpus. However, in Dutch also the synonym *lokale* of *plaatselijke* is used. The multilingual Dutch-to-English retrieval system will therefore find more correct fragments than the monolingual Dutch system.

Dutch morphology

Another volunteer also wanted to know something about the local Agenda 21 and formulated the query (*wat gebeurt er op*) *plaatselijk niveau met Agenda 21*, which was translated to *local level with Agenda 21*. The Dutch query of the first volunteer contains the inflected form *plaatselijke* of *plaatselijk*. Both are correctly translated to the same English word: *local*. Again, the multilingual Dutch-to-English retrieval system will find more correct fragments than the monolingual Dutch system.

Translational ambiguities

One of the volunteers formulated the query *inheemse volkeren*, which is translated by the system to *indegous people*. However, in the Dutch corpus the term *inheemse bevolking* is also used as a translation of *indegous people*. The meaning of *volkeren* and *bevolking* in Dutch is slightly different, the first meaning '*nation of people*' the second meaning '*population of people*'. Again, the multilingual Dutch-to-English retrieval system will therefore find more correct fragments than the monolingual Dutch system.

7.2.5 Conclusion

Using a very simple translation model we were able to built a simple but effective multilingual document retrieval system. It seems that the multilingual Dutch-to-English retrieval systems performs better than the monolingual Dutch system, but... looks can be deceiving.

- Partially correct translations, because of the existence of Dutch compounds, lead to better recall measures, partially because of *the limitation of our domain*.
- The *simplicity of our retrieval system* is a bigger disadvantage for the Dutch language than for the English language. Including morphological analysis will improve recall of the monolingual Dutch retrieval system.
- Dutch synonyms and English to Dutch translational ambiguities are an advantage in a Dutch-to English multilingual retrieval system, but will probably negatively influence the performance of an English-to-Dutch retrieval system.

Chapter 8

Conclusions

In the introduction of this paper we formulated two research questions: In which way can statistical methods applied to bilingual corpora be used to create the bilingual dictionary? What can be said about the performance of the created bilingual dictionary in a multilingual IR system? In this paper we built a system that is able to generate a bilingual dictionary from parallel English and Dutch texts. We tested the dictionary in a bilingual retrieval environment and compared recall and precision of a monolingual Dutch retrieval system to recall and precision of a bilingual Dutch-to-English retrieval system.

8.1 Building the dictionary

The proposed method for generating the bilingual dictionary uses the EM-algorithm to estimate the unknown parameters of a statistical translation model. We followed a new approach as we tried to generate a bi-directional dictionary. A bi-directional dictionary will save a lot of space in a multilingual retrieval environment. Because of the bi-directional approach we cannot use algorithms and models developed at other research centra.

1. we constructed a symmetric language model with independent source and target language words
2. we replaced the calculation of the E-step by the Iterative Proportional Fitting Procedure (IPFP)

We have no theoretical proof of the correctness our EM-algorithm in combination with the IPFP. However, comparison of different EM-algorithms indicates that our bi-directional approach may very well lead to better estimates than the unidirectional approach. More research is therefore needed on symmetric language models and estimating algorithms.

The current implementation shows promising results on the Agenda 21 corpus. The Agenda 21 corpus is probably too small to make good statistical inference. Evaluations on for example the Canadian Hansard corpus, which is 500 times bigger than Agenda 21 may bring more clearness in the performance of the algorithm.

8.2 The bilingual retrieval performance

We tested the usefulness of the generated bilingual dictionary in a multilingual retrieval environment. It seemed that the multilingual retrieval system performed better than the monolingual system. Dutch queries that were automatically translated by the retrieval system to English were able to retrieve 82% of the known relevant English fragments. The retrieved documents had a precision of 67%. The Dutch queries themselves were able to find 51% of the known relevant documents, but the documents were retrieved with a precision of 78%.

To explain the good performance of the multilingual retrieval system, we assigned each English query to one of the categories *correct*, *usable*, *partially usable*, *not translated* and *incorrect*. It seemed that 54% of the English queries fell into the first two categories, which represents by our criteria a reasonable translation. Only 15% of the English queries fell into the incorrect category. Still these figures do not explain the relatively good performance of the multilingual retrieval system, compared to the monolingual system. There are three important reasons for the unexpected good performance of the multilingual retrieval system (see Appendix B for explanation of the linguistic terms).

1. The simplicity of our retrieval system is a bigger disadvantage for the Dutch language than for the English language. Including *morphological analysis* will improve recall of the monolingual Dutch retrieval system.
2. Our translation does not account for the existence of Dutch *compounds* that ought to be translated to two separate English words. Partially correct translations, because of the existence of Dutch compounds, lead to better recall measures, because of *the limitation of our domain*.
3. Dutch *synonyms* and English-to-Dutch *translational ambiguities* are an advantage in a Dutch-to-English multilingual retrieval system as they improve recall retaining high precision measures. However, the same linguistic phenomena will probably negatively influence the performance of an multilingual English-to-Dutch retrieval system.

The problems mentioned under 1. and 3. are fundamental (linguistic) problems of monolingual Dutch document retrieval. Even if the translation system translates all Dutch queries correctly to English, these linguistic phenomena will cause different performance measures. The multilingual system makes them visible, but they are also there if a Dutch user wants to retrieve something from a Dutch database. Due to the richer Dutch morphology and due to the more frequent use of synonyms in Dutch (compared to English, if the Agenda 21 corpus is a good indication) Dutch information retrieval cannot achieve the same performance as English information retrieval. Dutch information retrieval may very well be a more difficult task than English information retrieval.

Basic information retrieval (IR) processes are *query formulation* (representing the information need), *indexing* (representing the documents), *retrieval* (comparing these representations) and *relevance feedback* (evaluating the retrieved documents). As each language has its own unique problems, techniques used for these processes will be different for each of them. Most research on document retrieval has been done using English. Techniques developed for English databases may need to be different if other languages are used. A major issue in multi-lingual information retrieval is whether techniques that have shown to be effective on English can be transferred to other languages, or how they have to be modified to achieve the best performance. More research is needed therefore on the performance of information retrieval techniques with different languages.

8.3 Recommendations to improve the translation system

In this paper we used very simple techniques to define equivalence classes. We used a relatively simple translation model. We used a relatively small corpus to estimate the parameters of the model. We did not use the possibility of estimating separate language probabilities $P(E)$ and $P(D)$. All of these steps can be easily improved. A lot of techniques to define equivalence classes, enhance the translation model and estimate language probabilities are already documented by other research centra (see chapter 3).

8.3.1 Equivalence classes

Defining equivalence classes of words using Morphological analysis may help the system in two ways. The number of entries of both dictionaries will be reduced. The EM-algorithm will be able to find correspondences a full word-based approach will overlook.

8.3.2 The translation model

To build a serious translation system with our bilingual dictionary, we first have to define a more realistic model of the prior probability. A bigram model will give the translation system the possibility to take the context of words into account when translating. Research also has to be made into more realistic models of the channel probability. Position probabilities account for the observation that words in the beginning of a Dutch sentence are most likely to be translated to the beginning of the English sentence. Fertility probabilities account for the observation that a word may be translated to more than one word (see chapter 3).

We have limited ourselves in this paper to English-Dutch translations. Because we did not use any knowledge of the languages we modelled we believe that the techniques we used will be equally useful on German and French; the other languages of the Twenty-One project. In a true multilingual translation system it must be relatively easy to add a language to the system. However, if we for example try to add French to our translation system we have to find a French-English corpus and a French-Dutch corpus. Research has to be done if the compilation of, for example a French-Dutch dictionary can benefit from the existence of an English-Dutch dictionary and an English-French dictionary.

8.3.3 Improving the parameter estimation

To improve the estimation of the channel probability we might want to use other resources than bilingual corpora. Research can be made into the use of human dictionaries and MT-lexicons to enhance the estimation of the channel probability.
Because the translation system is used in an information retrieval environment we can use the contents of the document base to adapt the prior probability (i.e. the language model). Each time a new document is added to the document base, we can adapt our language model, making the translation system 'aware' of the fact that a number of new subjects are added.

References

[Arnold, 1994] D. Arnold et al., *Machine translation: an introductory guide*, Blackwell Publishers 1994

[Brown, 1990] P.F. Brown, J. C. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, P.S. Roossin, *A Statistcal Approach to Machine Translation*, Computational Linguistics 16(2), 1990, pag. 79-85

[Brown, 1991] P.F. Brown, J.C. Lai, R.L. Mercer, *Aligning sentences in parallel corpora*, Proceedings of the 29th Annual Meeting for the Association for Computational Linguistics, Berkeley CA, 1991, pag. 169-176

[Brown, 1993] P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, R.L. Mercer, *The Matchematics of Statistical Machine Translation: Parameter Estimation*, Computational Linguistics 19(2), 1993, pag. 263-313

[Chen, 1993], S.F. Chen, *Aligning Sentences in Bilingual Corpora using Lexical Information*, Proceedings of the 31st Annual Meeting of the Association of Computational Linguistics, 1993, pag 9-16

[Chomsky, 1957] N. Chomsky, *Syntactic Structures*, Mouton, 1957

[Chomsky, 1965] N. Chomsky, *Aspects of the Theory of Syntax*, MIT press. Cambridge, 1965

[Church, 1991] K.W. Church and W.A. Gale, *A comparison of the enhanced good-turing and deleted estimation methods for estimating probabilities of English bigrams*. Computer Speech and Language 5, 1991, pag. 19-54

[Church, 1993] K.W. Church and R.L. Mercer, *Introduction to the Special Issue on Computational Linguistics Using Large Corpora*, Computational Linguistics 19(1), 1993, pag 1-24

[Croft, 1995] W.B. Croft and J. Xu, *Corpus-specific stemming using words from co-occurrence*, Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval, 1995, pag. 147-159

[Croft, 1996] W.B. Croft, J. Broglio and H. Fuji, *Applications of Multilingual Text Retrieval*, University of Massachusetts, 1996

[Dempster, 1977] A.P. Dempster, N.M. Laird and D.B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm* plus *discussions on the paper*, Journal of the Royal Statistical Society 39(B), 1977, pag 1-38

[v/d Eijk, 1993] P. van der Eijk, *Automating the Acquisition of Bilingual Terminology*, In the Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics, 1993, pag 113-119

[Everitt, 1992] B.S Everitt, *The Analysis of Contingency Tables*, Second edition, Chapman & Hall, London, 1992

[Gale, 1991] W.A. Gale and K.W. Church, *Identifying Word Correspondences in Parallel Texts*, Fourth DARPA Workshop on Speech and Natural Language, Morgan Kaufmann Publishers inc., 1991, pag 152-157

[Gale, 1993] W.A. Gale and K.W. Church, *A Program for Aligning Sentences in Bilingual Corpora*, Computational Linguistics 19(1), 1993, pag 75-102

[Gent, 1996] J. van Gent, W. Kraaij, R. Ekkelenkamp, J. den Hartog, *The Twenty-One Demonstrator Global Functional Design*, 1996 (draft version)

[Hemels, 1994] F. Hemels, *Reminder*, Twente University Master's Thesis, 1994, pag 15-36

[Hutchins, 1992] W.J. Hutchins, et al., *An introduction to Machine Translation*, Academic Press, 1992

[Kay, 1993] M. Kay and M. Röscheisen, *Text-translation alignment*, Xerox Palo Alto Research Center, 1993

[Klavans, 1990] J.L. Klavans and E. Tzoukermann, *The BICORD system, combining lexical information from bilingual corpora and machine readable dictionaries*, Proceedings of the 13th Annual Meeting of the Association of Computational Linguistics, 1990, pag 174-179

[Klavans, 1995] J.L. Klavans and E. Tzoukermann, *Combining Corpus and Machine-Readable Dictionary Data for Building Bilingual Lexicons*, Machine Translation 10(2), pag 185-218

[Kraaij, 1994] W. Kraaij and R. Pohlmann, *Porter's stemming algorithm for Dutch*, Research Institute for Language and Speech Utrecht University, 1994

[Krenn, 1996] B. Krenn and C. Samuelsson, *The Linguist's Guide to Statistics: don't panic*, Universität des Saarlandes, Saarbrücken, 1996 (draft version)

[Kupiec, 1993] J. Kupiec, *An Algorithm for finding Noun Phrase Correspondances in Bilingual Corpora*, Proceedings of the 31st Annual Meeting of the Association of Computational Linguistics, Columbus, Ohio, 1990, pag 17-22

[Kumano, 1994] A. Kumano and H. Hirakawa, *Building a MT Dictionary from Parallel Texts based on Linguistic and Statistical Information*, Proceedings of th 15th COLING, 1994, pag 76-81

[Mauldin, 1991] M.L. Mauldin, *Conceptual Information Retrieval: A case study in adaptive partial parsing*, Kluwer Academic Publishers 1991

[Mood, 1963] A. M. Mood and F.A. Graybill, *Introduction to the Theory of Statistics*, Second edition, McGraw-Hill Book Company, inc., Japan, 1963

[Pissanetzky, 1984] S. Pissanetzky, *Sparse Matrix Technology*, Academic Pres inc., London, 1984, pag. 4-37

[Porter, 1980] M.F. Porter, *An algorithm for suffix stripping*, Program 14(3), 1980, pag 130-137

[Shannon, 1948] C.E. Shannon, *A mathematical theory of communication*, Bell Systems Technical Journal 27, 1948, pag 379-423, 623-656

[Shannon, 1951] C.E. Shannon, *Prediction and entropy of printed English*, Bell Systems Technical Journal 30, 1951, pag 50-64

[Smadja, 1996] F. Samdja, K.R. McKeown, V. Hatzivassiloglou, *Translating Collocations for Bilingual Lexicons: A Statistical Approach*, Computational Linguistics 22(1), 1996, page 1-38

[Utsuro, 1994], T. Utsuro et al., *Bilingual Text Matching using Bilingual Dictionary and Statistics*, Proceedings of the 15th COLING, 1994, pag 1077-1082

[Wu, 1995] D.Wu and X. Xia, Large-Scale Automatic Extraction of an English-Chinese Translation Lexicon, Machine Translation 9, 1995, pag 285-313.

Appendix A

Elementary probability theory

This appendix briefly sketches the essentials of the probability theory. The definitions and notations introduced in appendix A are used throughout this paper. The appendix is based mainly on [Mood, 1963] and [Krenn, 1996].

A.1 The axiomatic development of probability

The notion of the likelihood of something is formalised through the concept of an *experiment*. An experiment is the process by which an observation is made. We assume a collection of basic outcomes for the experiment, which is called the *sample space* Ω. Let an event A be a subset of Ω. Then P will be called the *probability function* or *probability measure* on the sample space Ω if the following three axioms are satisfied.

$$\begin{aligned} &P(A) \geq 0 \ \textit{for every event } A \textit{ in } \Omega \\ &P(\Omega) = 1 \\ &\textit{If } A_1, A_2, \ldots \textit{ is a sequence of mutually exclusive events in } \Omega, \\ &\quad \textit{then } P(A_1 \cup A_2 \cup \ldots) = P(A_1) + P(A_2) + \ldots \end{aligned} \tag{1}$$

We call $P(A)$ the probability of the event A. A *probability space* is defined by a sample space Ω and a probability measure P. Because we are working with linguistic data, we will usually deal with discrete sample spaces Ω which contain a finite number of basic outcomes.

A.2 Conditional probability and Bayesian Inversion

If we have partial prior knowledge about the outcome of an experiment we capture this knowledge through the notion of *conditional probability*. The conditional probability of an event $A \subset \Omega$ given that an event $B \subset \Omega$ has occurred is:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad , P(B) > 0 \tag{2}$$

Bayes' inversion formula is a trivial consequence of the definition of conditional probability.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad , P(B) > 0 \tag{3}$$

The formula will be used in the following context. If we are looking for the event A that maximises $P(A|B)$ (for example that sentence A in English, that is the most probable translation of a given Dutch sentence B) we might as well look for the event A that maximises $P(B|A)P(A)$. This because $P(B)$ is constant, because it is give that B happend. This allows us to use seperate probability measures $P(A)$ of the sentences and $P(B|A)$ of the translations.

A.3 Random variables

A discrete *random variable X* is a function $X : \Omega \rightarrow N$. It allows us to reason with the probabilities of numerical values that are related to event spaces. If ω is a point in the sample space Ω, then $X(\omega) = x$ is the value of the random variable *X*. The *probability distribution* of a random variable *X* is simply defined by

$$P(X = x) = P(A_x), \quad A_x = \{\omega \in \Omega \mid X(\omega) = x\} \tag{4}$$

A.4 Expectation of a random variable

The expectation is the mean or average of a random variable. If *X* is a random variable with a probability distribution $P(X = x)$ then the expectation is defined by

$$E(X) = \sum_x x \cdot P(X = x) \tag{5}$$

unless the summation is unbounded.

A.5 Joint, marginal and conditional distributions

Suppose Ω is a sample space on which a probability measure *P* is defined. If we define two random variables *X* and *Y* over Ω, then the pair (*X, Y*) is called a two dimensional random variable and the two random variables *X* en *Y* are said to be *jointly distributed*. The joint distribution is denoted by

$$P(X = x, Y = y) \tag{6}$$

Related to a joint distribution $P(X = x, Y = y)$ are *marginal distributions* $P(X = x)$ and $P(Y = y)$ which are defined by

$$P(X = x) = \sum_y P(X = x, Y = y) \quad \text{and} \quad P(Y = y) = \sum_x P(X = x, Y = y) \tag{7}$$

We define the *conditional distribution* similar to the conditional probability as

$$P(X = x \mid Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}, \quad P(Y = y) > 0 \tag{8}$$

Note that the notation *P*(...) is overloaded. Any time we are talking about a different probability space, then we are talking about a different measure *P*. It is important to realise that one equation is often referring two several probability measures, all ambiguously referred to as *P*.

Appendix B

Linguistic phenomena

In this appendix we explain the linguistic terms used in this paper. We will deal with *morphology* , *ambiguity* , *collocations* and *idiom.*

B.1 Morphology

Morphology is concerned with internal structure of words and with how words can be analysed and generated. Morphological analysis is often omitted in language technology systems, because they are time consuming compared with full-form dictionary look up, or because of the high cost for developing algorithms. However, including a model of morphological analysis has a number of advantages of which we will name three in this paragraph. We will also mention the three main areas of morphology.

B.1.1 Advantages of morphological analysis

First, it is often possible to reduce the size of dictionaries considerably. Although English, for example, has a relatively impoverished inflectional morphology, the size of an machine translation (MT) lexicon can be almost halved by treating singular/plural alternations systematically and by limiting verb entries to root forms [Huthchins, 1992]. With the other Twenty-One languages (French and German besides English and Dutch) even greater savings can be derived.
The second advantage may be more important in this paper. It is the possibility for the system to spot relations between words that have the same basic meaning. If the objective is to analyse large corpora, morphological analysis can benefit from statistical regularities that a full word based approach must overlook [Brown, 1990].
The third advantage is the possibility to handle unknown words in a correct way. From the identification of grammatical inflections, it is often possible to infer syntactic functions even if the root is unknown [Huthchins, 1992].

B.1.2 Areas of morphology

Usually three different word formation processes are recognised: *inflection*, *derivation* and *compounding* [Arnold, 1994].

1. The inflectional process derives a word from another word form, maintaining the same syntactic category and basic meaning (e.g. father → fathers);
2. the derivational process derives a word from another word form, changing the syntactic category (e.g. father → fatherly);
3. the compounding process derives a word from two independent words to form a new unit (e.g. grand, father → grandfather).

In English inflection and derivation usually involve prefixes (e.g. do → undo) and suffixes (e.g. stupid → stupidity). In other languages like Dutch, a range of devices such as changes of consonants and duplicating vowels of words also are found (e.g. gave → gaaf) [Kraaij, 1994].

B.2 Lexical ambiguity

The problem of lexical ambiguity occurs if there are potentially two or more ways in which a word can be analysed. Lexical ambiguities are of three basic types: *category ambiguities*, *homographs and polysemes*, and *translational ambiguities* [Huthchins, 1992].

1. Category ambiguity occurs if a given word may be assigned to more than one grammatical or syntactic category (e.g. light can be a noun, verb or adjective).
2. Homography and polysemy occur when a word can have two or more different meanings. Homographs are words with quite different meanings, which have the same spelling. (e.g. a bank is a riverside and a financial institution). Polysemes are words which exhibit a range of meanings related in some way to each other. (e.g. mouth can also be used in 'mouth of a river'). In MT analysis homography and polysemy are often treated alike.
3. Translational ambiguities arise when a single source language word can potentially be translated by a number a different target language words. The source language word itself is not ambiguous, or rather it is not perceived by native speakers of the language to be ambiguous (e.g. the English 'wall' can in Dutch be 'wand', if inside a building or 'muur' if outside). Translational ambiguities occur more often if languages are not related (like English and Japanese) and may cause less problems in Twenty-One.

The problems with lexical ambiguity can be resolved by looking at the context. Two special cases of lexical ambiguous words and their context are considered in the next paragraph.

B.3 Other ambiguity problems

Whereas lexical ambiguities involve problems of analysing individual words and transferring their meanings, ambiguity problems with the syntactic structures and representations of sentences are also quite common. These problems include *structural ambiguity*, *anaphora ambiguity*, and *quantifier scope ambiguity* [Huthchins, 1992]. Because in Twenty-One only simple syntactic structures like noun phrases have to be translated, these types of ambiguity are not of the greatest concern in this paper.

B.4 Collocations and idiom

A *collocation* is a multiword unit of which the meaning can be understood from the meanings of the single words, but the particular words used are not predictable [Arnold, 1994]. Unlike the single words, the syntactic unit itself is not ambiguous and often can be translated to only one unit in the target language. In English, for example, translating text with a computer is referred to as 'machine translation', but not as 'automatic translation'. In French however 'traduction automatique' is the common term. When analysing a large bilingual corpus to find translations of single words, we will find 'automatique' as a possible translation of 'machine'.

In this paper *idiom* will be defined as a multiword unit of which the meaning cannot be understood from the meanings of the single words [Arnold, 1994]. Often idioms are ambiguous in the source language. In English, for example, 'to kick the bucket' usually means 'to die', but it is possible that the phrase really is about buckets and kicking. When analysing a bilingual corpus, idioms cause problems that are hard to overcome.

B.5 Structural differences

Suppose we didn't have any of the ambiguity problems mentioned above. Even then we would still be faced with difficult translation problems. Some of these problems are to do with lexical differences in the ways in which languages seem to classify the world, what concepts they choose to express by single words, and which they choose not to lexicalize. A particularly obvious example of this involves *lexical holes*, that is cases where one language has to use a phrase to express what another language expresses in a single word. For example the French *ignorer* has the English equivalent *to not know* or *to be ignorant of*.

f

Appendix C

Statistical Estimators

In this appendix we will explore different statistical estimators. In this paper we used simple Maximum Liklihood Estimatoin (MLE) to find a good statistical estimator to estimate the parameters of the translation model. However, MLE is not always a good statistical estimator because it assign zero probability to events that were not observed. Statistical estimators that assign some probability to unknown events are often used to design a more robust system.

A comparison of the statistical estimators introduced in this appendix was made for an English bi-gram model by Church and Gale [Church, 1991]. This appendix is mainly based on their article.

C.1 Introduction

In order to predict both the English to Dutch dictionary $P(D|E)$ and the Dutch to English dictionary $P(E|D)$ we only have to estimate the probabilities of the joint distribution $P(E,D)$. The conditional distributions follow by definition from the joint distribution $P(E,D)$, since

$$P(D|E) = \frac{P(E,D)}{P(E)} \quad \text{and} \quad P(E|D) = \frac{P(E,D)}{P(D)} \tag{1}$$

Suppose we posses a parallel corpus which we call the training text, that consists of N training instances; suppose a training instance is defined by a pair of words (E_i,D_i) that are each others translation (so we assume that we have 'complete data'). Let B be the number of equivalence classes (or bins) training instances are divided into. If we assume that there are r different English words and c different Dutch words then $B = rc$. Let $f(E_i,D_i)$ be the frequency of a certain translation pair in the training text. Let us say that there are N_k translation pairs that appeared k times in the training text.

N	Number of training instances consisting of translation pairs (E_i,D_i)
B	Number of equivalence classes training instances are devided into
$f(E_i,D_i) = k$	Frequency of an translation pair (E_i,D_i)
N_k	Number of equivalence classes that have k training instances in them

C.2 Maximum likelihood estimation

Regardless of how we form equivalence classes, we will end up with classes that contain a certain number of training instances. Suppose we found 10 instances of the English word *additional* in the English part of a certain corpus and of those, 8 were translated by the Dutch word *aanvullende*, once by *additionele* and once by *extra*. The question at this point is what

probability estimate we should use for estimating the Dutch translation of the English *additional*. The obvious answer for estimating the conditional probability is

$$P(aanvullende \mid additional) = 0.8$$
$$P(additionele \mid additional) = 0.1$$
$$P(extra \mid additional) = 0.1$$
$$P(\omega \mid additional) = 0.0, \text{ for } \omega \text{ not among the above three Dutch words} \quad (2)$$

These estimates are called maximum likelihood estimates (MLE). As we have already seen in chapter 3 the MLE makes the training data as probable as possible. It does not waste any probability mass on events that are not in the training corpus.

$$P_{\text{MLE}}(D|E) = \frac{f(E,D)}{f(E)} \quad \text{and} \quad P_{\text{MLE}}(E,D) = \frac{f(E,D)}{N} \quad (3)$$

However, the MLE is not always a suitable estimator. The problem is the sparseness of our data. While a few words are common, the vast majority of words are very uncommon. Valid translations of these words may very well be present in the corpus, but not all of them actually as a translation pair. The probability of a long English- Dutch sentence pair $P(E,D)$ is computed by multiplying the probabilities $P(E_i,D_i)$ that the words are translations. Because the MLE assigns zero probability to unseen events, one word pair in the test sentence, that was previously unseen in the training data, will give us bad (zero probability) estimates for the probability of valid sentences.

This problem is very notorious for *n*-gram estimations and will probably cause lesser problems if we are estimating the channel probability $P(D|E)$. This because we expect the average number of possible words that can follow another word to be much more then the average number of possible translations of a word. Nevertheless, we may still want to investigate some other estimators. A more formal approach to MLE is given in chapter 6.

C.3 Laplace's Law

For reasons stated in the previous chapter we may want to decrease the probability of previously seen events somewhat, so that there is a little bit of probability left over for previously unseen events. The oldest solution (1775) is to employ Laplace's law.

$$P_{\text{Lap}}(E,D) = \frac{f(E,D)+1}{N+B} \quad (4)$$

Note that the estimates Laplace gives are dependent on the number of equivalence classes *B*. For sparse training data over large English and Dutch vocabularies Laplace's law gives far too much of the probability space to unseen events.

C.4 Held out estimation

How can we know that too much probability is given to unseen events. One way we can test this is empirically. We can take further text (preferably from the same source) of the same length as the training text and see how often translations (English-Dutch word pairs) that appear *k* times in the training text tend to turn up in the further text. Suppose C_k is the total number of times that all translations that appeared *k* times in the training text appeared in the further text. Then the average frequency of those translations is C_k / N_k. The held out estimation is given by:

$$P_{\text{ho}}(E,D) = \frac{C_k}{N_k N}, \quad \text{where } k = f(E,D) \tag{5}$$

C.5 Deleted estimation

In order to derive the held out estimation we have to held out a part of the original training data. But if we do so, we actually use less training data and so our probability estimates will be less accurate. This is a common pattern in Statistics, where one ends up needing three pots of data: the basic training data, additional training data to smooth the initial probability estimates, and finally test data in order to evaluate the system. Rather than using some of the training data only for frequency counts, and some only for smoothing, a more efficient scheme is possible. We can divide the training data in two parts and use each part both as initial training data and as held out data. These methods are known as *cross-validation* methods.
Let $N_k^{(1)}$ be the number of translation pairs occurring k times in the first part of the training data and $C_k^{(1)}$ be the total occurrences of those bigrams in the other part. Let $N_k^{(2)}$ and $C_k^{(2)}$ be the same totals of the second part. Then deleted estimation is defined by

$$P_{\text{del}}(E,D) = \frac{C_k^{(1)} + C_k^{(2)}}{N(N_k^{(1)} + N_k^{(2)})}, \quad \text{where } k = f(E,D) \tag{6}$$

C.6 Good-Turing estimation

Good-Turing estimation is a method for determining probability estimates on the assumption that their distribution is binomial. The probability estimates of the form $P_{\text{GT}} = k^* / N$, where k^* can be thought of as the adjusted frequency. This frequency is adjusted according to the hypothesis that for previously observed items:

$$k^* = (k+1)\frac{E(N_{k+1})}{E(N_k)} \tag{7}$$

where E denotes the expectation of a random variable. The total probability reserved for unseen events is $E(N_1) / N$. Since the estimates for high values of k will be unreliable (the adjusted frequency of the maximum frequency k in the training data will be adjusted to zero) some curve S is used to smooth the values N_k. This leads to a family of possibilities for which

$$P_{\text{GT}}(E,D) = \frac{k^*}{N}, \quad \text{where } k^* = \frac{(k+1)S(N_{k+1})}{S(N_k)}, \quad r = f(E,D) \tag{8}$$

Depending on the curve S different Good Turing estimators can be defined, for instance Enhanced Good Turing.

C.7 Comparison of Statistical Estimators

In this paragraph we consider some data discussed by Church and Gale [Church, 1991] in the context of their discussion of various estimators for English bi-grams. Their corpus of 44 million words of Associated Press newswire yielded a vocabulary of 400,653 words (they maintained case distinctions, splitting on hyphens, etc.). This means there were 1.6×10^{11} possible bigrams, so a priori barely any of them will actually occur in the corpus. Church and Gale used half the corpus as a training text. An 'empirically determined gold standard' was estimated with the held out estimator allowing access to the other 22 million words. The other estimates are calculated only form the 22 million words of training data. The frequency

estimates that they derive are shown in table C.1. Probability estimates can be derived by dividing the frequency estimates by 22 million which is the number of bigrams.

$k = f_{MLE}$	$f_{empirical}$	f_{Lap}	f_{del}	f_{GT}	N_k	C_k
0	0.0000270	0.000137	0.0000374	0.0000270	74,671,100,000	2,019,187
1	0.448	0.000274	0.396	0.446	2,018,046	903,206
2	1.25	0.000411	1.24	1.26	449,721	564,153
3	2.24	0.000548	2.23	2.24	188,933	424,015
4	3.23	0.000685	3.22	3.24	105,668	341,099
5	4.21	0.000822	4.22	4.22	68,379	287,776
6	5.23	0.000959	5.20	5.19	48,190	251,951
7	6.21	0.001096	6.21	6.21	35,709	221,693
8	7.21	0.001233	7.18	7.24	27,710	199,779
9	8.26	0.001370	8.18	8.25	22,280	183,971

Table C.1: Estimates from Church and Gale

By comparing the three derived frequencies with the empirical derived frequency $f_{empirical}$ (which is derived by held out estimation using the test data) we can evaluate the different estimators. The Laplace estimator obviously gives a very poor estimate because it gives far to much probability to the unseen events (of which there are very many). Deleted estimation produces results that are quite close to the empirical derived estimate, but it nevertheless overestimates the expected frequency of unseen object considerably, by underestimating the objects that were seen once in the training data. Finally, the Good-Turing estimator gives exceedingly good estimations.

Appendix D

The Implementation

In this appendix we will briefly mention some technical details of the experiment that was conducted on the Agenda 21 corpus.

D.1 Sentence identification

Heuristic procedures were implemented using the Microsoft Word (MS Word) processor 6.0 for Windows on a regular Personal Computer. MS Word allows the user to define Find and Replace commands in a macro language. The sentence identification task was implemented in four different stages.

1. Identifying abbreviations and special characters (like for instance '%' and '$') and replacing them by their full words. This stage is a language dependent stage (that is, it differs per language).
2. Identifying paragraph headings by their numbers and marking them. This stage is language independent. Because the Agenda 21 contains a lot of (sub-)headings, marking this headings allows us to identify parallel paragraphs almost without error.
3. Identifying numerical expressions and replacing them with a special token. This stage is independent of the language
4. Identifying sentences by points, question marks, etc. and marking them. This stage is independent of the language

For both English and Dutch, hyphens were replaced by spaces. Splitting words at the and of a line never occurred in the parallel corpus. In Dutch diacritics on characters were removed. We made no distinction between capitals and non-capitals

D.2 Sentence alignment

The sentence alignment task was done using an implementation of Church and Gale [Church, 1993] and carried out on a Unix work station. The program makes use of the fact that longer sentences in one language tend to be translated into longer sentences in the other language, and shorter sentences tend to be translated into shorter sentences. A distance measure of the character length of a sentence is used as a criteria.

The program inputs two text files, with one token (word) per line. The text files contain a number of delimiter tokens. There are two types of delimiter tokens: "hard" and "soft." These delimiter tokens are added in the sentence identification step of the process. The hard regions (e.g., paragraphs) may not be changed, and there must be equal numbers of them in the two input files. The soft regions (e.g., sentences) may be deleted (1-0), inserted (0-1), substituted (1-1), contracted (2-1), expanded (1-2), or merged (2-2) as necessary so that the output ends up with the same number of soft regions.

D.3 Word alignment

We implemented the EM-algorithm using the programming language C on a Unix work station. Sparse matrix technology [Pissanetzky, 1984] was used to implement a datastructure that uses minimal memory to hold two matrix copies with the same zero values. One copy is used for the probability estimates, the other is necessary to collect the frequency counts. The datastructure used is called a sparse row-wise ordered matrix format. Each row consists of a list of three words (a word consists of two bytes), the first contains the column index, the second and the third contain the values of both matrix copies. Two values with the same column indexes are allowed indicating that the value has to be stored in four bytes. A pointer list is needed to find the start of each row. Each matrix copy needs little more than 33% memory overhead, that is memory needed to find the right matrix cell. The program uses four stages to carry out the EM algorithm.

1. First the different words (or equivalence classes) are identified each word is assigned a numerical code. It is with these codes (that represent the words), that we are going to carry out the EM-algorithm. Sentences that exceed a maximum length are skipped.
2. The sparse matrix data structure with the initial estimates is built in this stage. This stage takes a lot of time, (about five hours on the Agenda 21 training corpus) but can be done more efficiently if better memory management is used.
3. This stage actually performs the EM-algorithm. Each step of the EM-algorithm takes about 10 minutes for the Agenda 21 corpus.
4. Finally the conditional probabilities can be computed so we can use both the English-Dutch and the Dutch-English dictionary.

D.4 The retrieval engine

The retrieval of the test fragments was simulated using MS Word 6.0 for Windows on a Personal Computer. A Boolean IR model was simulated with the *find* option of MS Word. In a boolean IR model the document is represented by a collection of *catch words*. A query consists of a list of catchwords seperated by the Boolean operators like AND, OR and NOT. Query and documents can be matched by checking if the catch words that belong to the documents will make the query true [Hemels, 1994].