Multilingual Domain Modeling in Twenty-One

Automatic Creation of a Bi-directional Translation Lexicon from a Parallel Corpus

Djoerd Hiemstra

University of Twente, Parlevink Group

Abstract

Within the project Twenty-One, which aims at effective dissemination of information on ecology and sustainable development, a system is developed that supports cross-language information retrieval for any of the four languages Dutch, English, French and German. Knowledge of this application domain is needed to enhance existing translation resources for the purpose of lexical disambiguation. This paper describes an algorithm for the automated acquisition of a translation lexicon from a parallel corpus. New about the presented algorithm is the statistical language model used. Because the algorithm is based on a symmetric translation model it becomes possible to identify one-to-many and many-to-one relations between words of a language pair. We claim that the presented method has two advantages over algorithms that have been published before. Firstly, because the translation model is more powerful, the resulting bilingual lexicon will be more accurate. Secondly, the resulting bilingual lexicon can be used to translate in both directions between a language pair. Different versions of the algorithm were evaluated on the Dutch and English version of the Agenda 21 corpus, which is a UN document on the application domain of sustainable development.

1 Introduction

Indexing large collections of documents that are written in various languages introduces special problems if the system has to support cross-language retrieval. In a cross-language retrieval system, the user can query the document base in the language of his/her choice to retrieve documents in any of the supported languages. The system has to perform some sort of automatic translation to support this functionality. General purpose dictionaries and MT-systems are not very well suited for this purpose, especially if the queries are short and the domain is restricted. A large domain-specific example text that is available in two languages can be used to extract words and translations of words that are common in the domain.

This paper describes an algorithm for the automatic extraction of a translation lexicon from parallel corpora. In contrast to earlier publications (D. Hiemstra 1997a, D. Hiemstra, F.M.G. de Jong, and W. Kraaij 1997b) this paper will introduce a new algorithm for detecting multi-word translations and give an extensive evaluation of the different algorithms and of the influence of preprocessing of the parallel corpus. The paper is organised as follows. Section 2 will give a brief outline of the context of the research presented in this paper: the project Twenty-One. Section 3 will describe previous work on the acquisition of bilingual lexicons from parallel corpora. The sections 4 and 5 introduce two probability models and corresponding algorithms. Finally, section 6 will present experimental results.

2 The project 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. In the project Twenty-One[1] environmental organisations, research organisations and companies work together to improve the distribution and use of common interest documents about ecology and sustainable development. Project partners are: DFKI, Xerox Grenoble, Getronics, TNO-TPD, University of Twente, University of Tübingen, MOOI foundation, Environ, Climate Alliance, VODO and Friends of the Earth.

2.1 The Twenty-One system

The most important deliverable of the project is a disclosure and retrieval system which produces a searchable index on a multilingual multimedia document base. This index will be available via cd-rom and the world wide web[2]. The Twenty-One document base consists of documents in different languages, initially Dutch, English, French and German but extensions to other European languages are envisaged. The system will support Cross-Language Information Retrieval (CLIR), meaning that users can query the document base in their favourite or native language and retrieve documents in any of the languages supported by the system. Documents will be (partially) translated before presenting them to the user making relevance judgements of the document possible (W. Kraaij 1997b, W.G. ter Stal, J-H Beijert, G. de Bruin, J. van Gent, F.M.G. de Jong, W. Kraaij, K. Netter, and G. Smart 1998).

2.2 Agenda 21

The name of the project Twenty-One refers to the United Nations conference on ecology and sustainable development in Rio de Janeiro in 1992. An important result of this conference is a document titled Agenda 21 which will serve as a test corpus in the experiments described in this paper. The reason for this choice is that Agenda 21 is a document on exactly the application domain targeted by the project. Furthermore, it is available in all the major European languages. For the final Twenty-One system six translation lexicons will be extracted for translation between any of the four languages Dutch, English, French and German. For the experiments described in this paper we will use the Dutch and English version of the Agenda 21 corpus. We are especially interested in experiments with Dutch as one of the languages because in Dutch compound nouns are written as one single word. The problem of mapping a Dutch compound noun consisting of one word to

[1] Twenty-One is a project funded by the European Union within the Telematics Applications Programme, sector Information Engineering.

[2] An intermediate version of the Twenty-One demonstrator is available on: `http://twentyone.tpd.tno.nl/`

an English compound noun consisting of several distinct words is one of the problems that will be addressed in this paper.

Agenda 21 consists of approximately 150.000 words in every language, which makes it a relatively small corpus compared to parallel corpora used in some recent publications on automatic lexicon acquisition. For instance Brown et al. (P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer 1993) used approximately 29 million parallel words from the English-French Canadian Hansards: about 200 times as much as the total size of our corpus. Getting hold of useful data that includes Dutch is more difficult. If we compare the size of the Agenda 21 corpus to parallel corpora used in publications that use Dutch as one of the languages, the corpus is actually relatively big. For instance Van der Eijk (P. van der Eijk 1993) used approximately 25.000 parallel words from the Dutch and English version of the official announcement of the ESPRIT programme: about one sixth of the total size of Agenda 21.

Although the Agenda 21 corpus is small it will be used to evaluate the method presented in this paper. We will compare the method's performance to the performance of an algorithm published by Brown et al. (P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer 1993) that already proved its usefulness on much bigger corpora. In the near future we also hope to evaluate the performance of the presented algorithm on a corpus that is bigger than Agenda 21.

3 Building a probabilistic bilingual lexicon

To model the domain of sustainable development a statistical algorithm will be applied on Agenda 21. The algorithm will generate a probabilistic bilingual lexicon. A probabilistic bilingual lexicon assigns to each possible translation of an entry a probability measure to indicate how likely the translation is. An example of the kind of entries the algorithm generates from the Dutch and English versions of Agenda 21 is given in Figure 1. A general purpose dictionary[3] or MT system[4] would translate the Dutch word *gevaarlijke* to the English word *dangerous*, in the domain of sustainable development, the most common English translations are *hazardous* and *toxic*.

Recently much research was done into statistical methods for the extraction of translation lexicons from parallel corpora. The first step in deriving a translation lexicon is finding the correspondences between sentences. For the sentence alignment problem well-documented solutions are available. We used an algorithm published by Gale and Church (W.A. Gale and K.W. Church 1993) that aligns sentences of a parallel corpus based on sentence lengths.

Roughly spoken two approaches can be taken to find the translations of the words within the sentences: the *hypothesis testing* approach and the *estimat-*

[3] The Van Dale translation dictionary Dutch-English (W. Martin and G.A.J. Tops et al., editors 1986) gives *dangerous* as the preferred translation of *gevaarlijke*. It also gives *hazardous*, but not *toxic*.

[4] Systran (Systran 1998) is not available yet for Dutch-English, but the French-English version will translate ...*déchets dangereux*... (within the proper context) into *dangerous waste*; in the corpus it is *hazardous wastes*.

gevaarlijke	
hazardous	0.74
toxic	0.20
dangerous	0.05
⋮	⋮

Figure 1: An example entry

ing approach. The disadvantage of the hypothesis testing approach (W.A. Gale and K.W. Church 1991, P. van der Eijk 1993, F. Smadja, K.R. McKeown, and V. Hatzivassiloglou 1996) is that a valid hypothesis can only be made if a certain minimum number of observations is available. Therefore only a limited amount of translation examples can be found with high accuracy. Following the estimating approach, it is possible to find the most probable translations for each example in the parallel corpus. A disadvantage of the algorithms used in recent publications (P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer 1993, J. Kupiec 1993, D. Wu and X. Xia 1995) is the use of directional translation models. For these algorithms the probabilities estimated from, say, an English-Dutch corpus will be different from the probabilities estimated from a Dutch-English corpus, even if the English and Dutch texts are exactly the same for both experiments. A nice characteristic of these models is the fact that one word may be translated into several words in the other language. This is particular important if one of the languages is a compounding language. In for example Dutch nouns may be compounded unrestrictedly to build new words. For example *volksgezondheid* which means *human health* is build from *volk* (i.e. *human*) and *gezondheid* (i.e. *health*). However, the algorithms presented in previous publications do not model the fact that the same thing might happen the other way around. For example the English *overdependence* would be *overmatige afhankelijkheid* in Dutch.

This paper will follow the estimating approach because it is more robust than the hypothesis testing approach. The approach presented in this paper will differ from other approaches in the use of a symmetric translation model. We claim that the presented method has two advantages.

(i) Because the translation model is more powerful, the resulting bilingual lexicon will be more accurate.

(ii) Because the translation model is symmetric, the resulting bilingual lexicon can be used to translate in either direction between a language pair. This will require less space than two uni-directional lexicons.

4 Assigning probabilities to translations

The problem of modelling the translation of sentences may, to some extend, be compared to problems in medicine and social sciences. In many of these studies a population 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 paper. Suppose we want to study the bilingual corpus of Figure 2 that consists of four pairs of English and Dutch sentences which are each other's translation. There are two multi-word expressions in the example corpus. In Dutch an infinitive does not need the particle *to*, so *to walk* is the translation of the Dutch word *lopen* in this example. The English word *love* is the translation of *houden van* in which *van* is a preposition.[5]

I am nice.	*Ik ben leuk.*
To walk is nice.	*lopen is leuk.*
I love you.	*Ik houd van jou.*
I love to walk.	*Ik houd van lopen.*

Figure 2: An example parallel corpus

Just like the physician has to diagnose the condition of the patient he examines ("what type of cancer does the patient have?"), we will assign each observation to an equivalence class. Between observations that fall into the same equivalence class there exists an equivalence relation, i.e. the observations share a certain property. Which equivalence classes we define depends on the information available and the information we are interested in. It is for example possible to identify words (P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer 1993, W.A. Gale and K.W. Church 1991), noun phrases (P. van der Eijk 1993, J. Kupiec 1993) or collocations (F. Smadja, K.R. McKeown, and V. Hatzivassiloglou 1996). If some form of morphological analysis is performed we might want to assign different words to the same equivalence class. For example the English words *is* and *am* of the corpus of Figure 2 share the same base form *to be*.

One of the things we will evaluate in this paper is the influence of the different equivalence classes on the process of word alignment. Different preprocessing steps like morphological stemming, part-of-speech tagging, compound splitting and noun phrase identification will define different equivalence classes. For the example corpus we assume that there will be no linguistic preprocessing. Without any preprocessing than the identification of words there are $r = 8$ different English words and $c = 8$ different Dutch words. This makes a total of 64 possible translations that will be displayed in a so-called contingency table (Table 1).

In the contingency table, each English word has an unique row index i and each Dutch word has an unique column index j. The cell frequencies n_{ij} in the table represent the number of times the English word i and the Dutch word j are each other's translation in the corpus. In terms of cell frequencies n_{ij} the marginal totals

[5] The detection of multi-word expressions containing non-content words like *to* and *van* is not very useful for the purpose of information retrieval and only serves as an example.

Table 1: Contingency table for the example corpus

	ik	ben	$leuk$	$lopen$	is	$houd$	van	jou	
I	n_{11}	n_{12}			$\cdots$			n_{1c}	$n_{1.}$
am	n_{21}							$:$	$n_{2.}$
$nice$	$:$							$:$	
to									
$walk$									
is									
$love$									
you	n_{r1}				$\cdots$			n_{rc}	$n_{r.}$
	$n_{.1}$	$n_{.2}$			$\cdots$			$n_{.c}$	$n_{..}$

and the overall total are given by:

$$(1) \qquad n_{i.} = \sum_{j=1}^{c} n_{ij}, \quad n_{.j} = \sum_{i=1}^{r} n_{ij}, \quad n_{..} = \sum_{i=1}^{r}\sum_{j=1}^{c} n_{ij}$$

Each cell frequency n_{ij} will be assigned an unknown probability parameter p_{ij} which is the probability that the English word i and the Dutch word j appear in the corpus as a translation pair. To define the probability measure P as a function of the frequencies n_{ij} and the parameters p_{ij} it is assumed that the translation pairs in a sentence pair are independent of each other. Furthermore we assume that there is no order between words or translation pairs of words. These assumptions lead to the following model of the probability measure P:

$$(2) \qquad P(N = \begin{bmatrix} n_{11} \cdots n_{1c} \\ : \qquad : \\ n_{r1} \cdots n_{rc} \end{bmatrix}) = \frac{n_{..}!}{n_{11}! \cdots n_{rc}!} \prod_{i=1}^{r}\prod_{j=1}^{c} p_{ij}^{n_{ij}}$$

Equation 2 is a variant of the well known multinomial distribution and its unknown parameters p_{ij} form the probabilistic bilingual lexicon we are looking for. The estimate $\hat{p}_{ij}$ of p_{ij} that makes the observations as likely as possible is given by

$$(3) \qquad \hat{p}_{ij} = \frac{n_{ij}}{n_{..}}$$

which is the maximum likelihood estimate of the unknown parameters.

5 The estimation algorithm

In the previous section a model of the probability distribution P was introduced. However, Equation 3 cannot be used directly to estimate the unknown parameters of the model, because we do not know the translation of the words in the parallel corpus. Only the translation of each sentence in the corpus can be observed. The observation of parallel sentences in the corpus can be viewed as incomplete data, i.e. the frequencies n_{ij} cannot be observed directly but only indirectly via the sentences that are each other's translation.

Table 2: Incomplete observation of (*I am nice., Ik ben leuk.*)

	ik	ben	$leuk$	$lopen$	is	$houd$	van	jou	
I	**?**	**?**	**?**	-	-	-	-	-	1
am	**?**	**?**	**?**	-	-	-	-	-	1
$nice$	**?**	**?**	**?**	-	-	-	-	-	1
to	-	-	-	-	-	-	-	-	0
$walk$	-	-	-	-	-	-	-	-	0
is	-	-	-	-	-	-	-	-	0
$love$	-	-	-	-	-	-	-	-	0
you	-	-	-	-	-	-	-	-	0
	1	1	1	0	0	0	0	0	3

Table 2 shows the problem of incomplete observations. For the sentence pair (*I am nice., Ik ben leuk.*) all cell frequencies are known to be zero except for the nine in the upper left part of the table. Cell frequencies that are zero are displayed as '-'.

5.1 The EM-algorithm

Dempster et al. (A.P. Dempster, N.M. Laird, and D.B. Rubin 1977) formulated the Expectation Maximisation-algorithm (EM-algorithm) for finding maximum likelihood estimates from incomplete data. From the definition of the EM-algorithm the following iterative solution can be constructed that estimates the unknown probabilities from the incomplete observations.

(i) Take an initial estimate of the probability parameters.

(ii) *Expectation-step*: For each sentence s, calculate the expected cell frequencies $n_{ij}^{(s)}$ given the words in the observed parallel sentence and the probability parameters. Add up $n_{ij}^{(s)}$ for each s to get n_{ij}.

(iii) *Maximisation-step*: Calculate new estimates of the probability parameters as defined by the maximum likelihood estimator of equation 3.

(iv) Repeat (ii) and (iii) until the probability parameters do not change significantly anymore.

The expectation step of the algorithm is a non-trivial one. It is necessary to specify how the incomplete data is related to the complete data. In the following chapter two mappings from complete data to incomplete data will be introduced. A mapping from complete data to incomplete data will be called the alignment model.

5.2 Two alignment models

Brown et al. (P.F. Brown, J.C. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin 1990) introduced the idea of an alignment between words as an object indicating which words in a parallel sentence are each other's translation. Instead of displaying alignments in the contingency table of Table 1 they can be shown graphically as in Figure 3 by drawing arrows between words. Expected cell frequencies of the contingency table that are higher than a certain treshold correspond with lines between the words in Figure 3 and will be called connections.

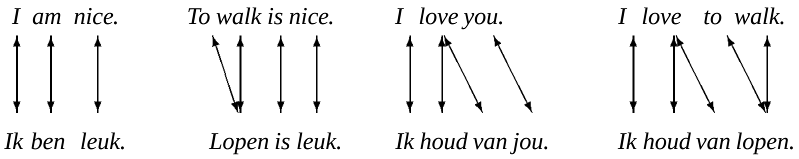

Figure 3: Graphical illustration of the aligned example corpus

We will define two symmetrical alignment models A and B. In model A words in both languages must have one and only one connection, so only one-to-one relations between words are allowed. In model B one-to-many and many-to-one connections are also allowed. Model A does only account for the first parallel sentence of the example corpus. The directional model introduced by Brown et al. accounts for two of the parallel sentences of the example corpus. Model B accounts for all four.

Model A

In model A it is assumed that every word in a sentence is translated to one and only one word in the other sentence. Because the sentences in the example corpus do not always have the same length, the assumption is made that some words are not translated at all. To model this assumption, a special $(null)$ word is introduced for each language. If the length of, for example, the English sentence is smaller than the length of the parallel Dutch sentence, the English sentence is filled up with the special $(null)$ words. Table 3 shows the expected cell frequencies of the example corpus of Figure 2 after the algorithm has converged. Because model A cannot model one-to-many and many-to-one translations, the result on the multi-word expressions is not satisfactory.

Model B

In model B one-to-many and many-to-one translations are allowed. This has some unfortunate implications on the probability function of Equation 2. Equation 2 has the property that the probabilities of all possible alignments sum up to one, given

Table 3: Expected frequencies after 5 iterations of model

	ik	*ben*	*leuk*	*lopen*	*is*	*houd*	*van*	*jou*	*(null)*	
I	3.0	-	-	-	-	-	-	-	-	3.0
am	-	1.0	-	-	-	-	-	-	-	1.0
nice	-	-	2.0	-	-	-	-	-	-	2.0
to	-	-	-	1.0	0.25	0.25	0.25	-	0.25	2.0
walk	-	-	-	1.0	0.25	0.25	0.25	-	0.25	2.0
is	-	-	-	-	0.5	-	-	-	0.5	1.0
love	-	-	-	-	-	1.0	1.0	-	-	2.0
you	-	-	-	-	-	0.25	0.25	0.5	-	1.0
(null)	-	-	-	-	-	0.25	0.25	0.5	-	1.0
	3.0	1.0	2.0	2.0	1.0	2.0	2.0	1.0	1.0	15.0

the number of connections $n_{\cdot\cdot}$ of the alignments. In model A the number of connections is known given the incomplete data[6] and it only makes sense to compare alignments that have the same value for $n_{\cdot\cdot}$.[7] In model B, however, the number of connections are unknown given the observation of a sentence pair. Therefore we would like to compare alignments that do not have the same number of connections and we will make the ad-hoc assumption that it *does* make sense to compare their probabilities according to Equation 2. More formally we assume that the probability of an alignment does not depend on the number of connections or the length of both sentences (we already made this assumption implicitly in Equation 2) and that the number of connections is uniformly distributed. One of the consequences of this assumption is that alignments with more connections generally will get a lower probability. This is probably a good thing as the algorithm will prefer one-to-one alignments over alignments that include a lot of multi-word expressions. Table 4 shows the expected cell frequencies after the algorithm has converged using model B on the example corpus.

Discussion

We did not give a formal definition of the mapping from complete to incomplete data. For model A the formal definition follows straightforwardly from the informal definition. For model B the formal definition is not that straightforward. The matrix N does not make a distinction between the translation of a multi-word expression and multiple occurrences of a word in a sentence. For model B the distinction is important. Therefore it is necessary to extend the matrix in such a way that every occurrence of a word gets its own row or column. In this way every marginal total bigger than one will refer to a multi-word translation.

[6] The number of connections is equal to the length of the longest sentence of a parallel sentence pair.

[7] For instance Hidden Markov Models also sum up to one given the number of state transitions and it usually does not make sense to compare the probabilities of two state sequences that do not share the same length.

Table 4: Expected frequencies after 5 iterations of model

	ik	ben	$leuk$	$lopen$	is	$houd$	van	jou	
I	3.0	-	-	-	-	-	-	-	3.0
am	-	1.0	-	-	-	-	-	-	1.0
$nice$	-	-	2.0	-	-	-	-	-	2.0
to	-	-	-	2.0	-	-	-	-	2.0
$walk$	-	-	-	2.0	-	-	-	-	2.0
is	-	-	-	-	1.0	-	-	-	1.0
$love$	-	-	-	-	-	2.0	2.0	-	4.0
you	-	-	-	-	-	-	-	1.0	1.0
	3.0	1.0	2.0	4.0	1.0	2.0	2.0	1.0	16.0

5.3 Calculating the E-step

Calculating the Expectation step of the EM-algorithm requires summing over all possible alignments of a sentence pair. For the example corpus of Figure 2 this is not a problem because the sentences are very small. However in a realistic corpus like Agenda 21 evaluating every single alignment cannot be done in reasonable time, because the number of possible alignments in a sentence pair increases exponentially with the length of both sentences. If, for example, l is the maximum length of both sentences, the number of possible alignments of model A is $l!$. As the average sentence length in our corpus is more than 20, the number of possible alignments is usually more than $20! > 10^{18}$. In this section different techniques to calculate the E-step of the EM-algorithm are introduced.

Generating functions

One of the most powerful devices for solving enumeration problems involves the use of so-called *combinatorial generating functions*. A generating function was used by Brown et al. (P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer 1993) to calculate the E-step of a directional alignment model. Their algorithm comes down to picking the most probable translation for every word in a sentence, without accounting for possible translations of other words in the same sentence. Both our models, however, are intractable and we know of no generating function that generates all combinations in polynomial time. We therefore turn our attention to methods that approximate the E-step.

Optimum path-finding algorithms

One strategy to approximate the E-step would be to search for the m most probable alignments. The A* algorithm (N.J. Nilsson, editor 1980) is a popular algorithm for finding an optimal path to a goal. The algorithm stores every trial path explicitly; any of them can be candidate for further extension. An evaluation function is used to determine which trial path has to be expanded. Search with A* will take much less calculation than computing all possible alignments, but the algorithm is still intractable. Nilsson (N.J. Nilsson, editor 1980) mentions a number of shortcuts to reduce the amount of computation of the search algorithm. If these shortcuts are used there is no guarantee that the solution is optimal. We experimented with optimum path-finding algorithms, but were not able to tune the algorithm in such a way that it gives reliable results in reasonable time.

Random Sampling

Another strategy to approximate the E-step of the EM-algorithm would be the process of random sampling. Suppose that, given an observed parallel sentence, alignments are generated in such a way that the chance of an alignment being selected is equal to its probability. If m alignments: $N^{(1)}, N^{(2)}, \cdots, N^{(m)}$ are generated by random sampling, then an estimator $\hat{n}_{ij}$ of the expected frequency n_{ij} would be:

$$\hat{n}_{ij} = \frac{1}{m} \sum_{k=1}^{m} n_{ij}^{(k)} \tag{4}$$

If we assume a connection to be binomially distributed with expectation μ then the theoretical standard error would be $\sigma = \sqrt{\mu(1-\mu)/m}$. In practice we do not know the standard error and we should estimate $\sigma = s$ from:

$$s^2 = \frac{1}{m-1} \sum_{k=1}^{m} (n_{ij}^{(k)} - \hat{n}_{ij})^2 \tag{5}$$

The theoretical standard error is inversely proportional to the square root of the sample size m. Therefore, to reduce the standard error by a factor of k, the sample size needs to be increased k^2-fold, meaning that it is possible to approximate the E-step with an arbitrarily small error in polynomial time. For the random sampling method which is often called Monte Carlo method we refer to Hammersley and Handscomb (J. Hammersley and D. Handscomb, editors 1964).

Iterative proportional fitting

The E-step of alignment model A can be approximated quick-and-dirty with standard matrix operations. The iterative proportional fitting procedure (IPFP) uses the fact that given the observation of a parallel sentence in model A, the marginal totals of the contingency table are fixed. IPFP takes a contingency table with initial frequencies $n_{ij}^{(0)}$ and iteratively scales the table to satisfy the observed marginal totals $m_{i.}$ and $m_{.j}$. The pth iteration of the algorithm consists of two steps which form:

$$\begin{aligned} n_{ij}^{(p,1)} &= n_{ij}^{(p-1,2)} \cdot m_{i.} / n_{i.}^{(p-1,2)} \\ n_{ij}^{(p,2)} &= n_{ij}^{(p,1)} \cdot m_{.j} / n_{.j}^{(p,1)} \end{aligned} \tag{6}$$

The first superscript refers to the iteration number, and the second to the step number within iterations. The algorithm continues until the observed data $m_{i.}$ and $m_{.j}$

and the marginal totals $n_{i.}^{(p)}$ and $n_{.j}^{(p)}$ are sufficiently close. We used the IPFP in an unconventional way with the following initial frequencies for each i and j:

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

Equation 7 is a relative risk approximation called odds ratio. It is based on the fact that initially $p_{i.} \gg p_{ij}$ and $p_{.j} \gg p_{ij}$. The marginal probability parameters $p_{i.}$ and $p_{.j}$ are defined according to the marginal totals in Equation 1. For IPFP and the odds ratio we refer to Everitt (B.S. Everitt, editor 1992).

6 Experimental Results

Algorithms for word alignment can be evaluated either over types or tokens (I.D. Melamed 1997a). We will test the algorithms over tokens, that is, by looking directly at the alignments and not at the resulting lexicon. For the experiments described in this section we used a training corpus from the English and Dutch version of Agenda 21 consisting of 5750 parallel sentences. Of these 5750 sentences, 20 sentences were aligned by hand. Some common closed class words like articles and prepositions were excluded from the annotation process. The test sentences were annotated with a total of 238 connections between word pairs. The test sentences were not excluded from the training corpus, because we were interested in the ability of the algorithms to align sentences in a training corpus. (Of course, the annotation of the test sentences was not included in the training corpus.) With the final evaluation of the Twenty-One cross-language retrieval system we will evaluate the probabilistic dictionaries on unseen data: a multilingual document collection.

6.1 Testing of algorithms

We implemented two versions of model A: one with random sampling approximation and one with IPFP approximation. For model B an algorithm using random sampling approximation was implemented. The results of the algorithms were evaluated as follows. After convergence, the E-step was calculated once more for every test sentence. Every expected frequency higher than or equal to 0.5 was considered a connection and was compared with the manually annotated data. Connections were classified as either *correct*, *wrong* or *missing* and the values for correct, wrong and missing were used to calculate measures that are standard in information retrieval literature: *precision*, *recall* and *F-measure*.

(8) $$precision = \frac{correct}{correct + wrong} \qquad recall = \frac{correct}{correct + missing}$$

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

We also implemented the IBM model 1 algorithm as published by Brown et al. (P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer 1993) which is a directional model. As this algorithm will produce different translation lexicons for

English-Dutch and Dutch-English, we did two separate experiments. The first experiment labeled with IBM NL→EN indicates the model that identifies multi-word expressions in Dutch. The second experiment labeled with IBM EN→NL indicates the model that identifies multi-word expressions in English.

Table 5: Evaluation of different algorithms

	model A, IPFP	model A, sampling	model B, sampling	IBM 1 NL→EN	IBM 1 EN→NL
precision	0.90	0.97	0.89	0.69	0.79
recall	0.74	0.69	0.70	0.67	0.76
F	0.81	0.81	0.78	0.68	0.77

Table 5 gives the results of the algorithms on the test sentences. It took our implementation of the IPFP version of model A about the same amount of time as our implementation of the IBM model 1 algorithm to align the corpus: about 20 minutes. The sampling versions are slower, it takes them about 2 hours to align the words of the parallel corpus.

Discussion

If the F-measure is taken as a criterion for the performance of the algorithms then both versions of model A outperform the other algorithms. The IPFP version of model A has a higher recall than the sampling version at the expense of its precision. This is probably due to the fact that the IPFP version converges completely, whereas the sampling version will keep on changing the parameters with an amount proportional to the standard error of the sampling algorithm.

Surprisingly the more sophisticated models perform worse. Model B performs little better than the EN→NL version of the IBM algorithm. The difference between the EN→NL version and the NL→EN of the IBM algorithm is quite dramatic, indicating the one should take care on how to use directional alignment algorithms for language pairs like Dutch and English.

6.2 Testing the influence of pre-processing

The results of the algorithms on the parallel corpus gives rise to the hypothesis that the performance of the model A algorithm can still be improved by intelligent preprocessing of the corpus. Lemmatisation can map morphologically related words into one equivalence class, which can reduce the probability space without introducing much extra ambiguity. A compound splitter for Dutch can free the algorithm from much of the multi-word translations of Dutch compound nouns. Another strategy to get around compound nouns in Dutch is noun phrase (NP) extraction in both Dutch and English.

We used the morphological tools by Xerox for lemmatisation, Part-of-Speech

(POS) disambiguation and compound splitting. We used a parser by TNO[8] to extract minimal NPs, i.e. without PP-attachments and embedded NPs. Words that were not part of a NP, were not removed and still had to be aligned by the algorithm. The IPFP version of model A was used to compare the effect of pre-processing methods, because it proved to be fast and pretty accurate. Table 6 gives the effect of the different pre-processing methods on the final results.

Table 6: Evaluation of the IPFP version of model with methods for pre-processing

	raw text	lemma plus POS	lemmatised	lemmatised comp. splitting	noun phrases
precision	0.90	0.91	0.89	0.93	0.88
recall	0.74	0.78	0.77	0.82	0.57
F	0.81	0.84	0.83	0.87	0.69

Discussion

Intelligent preprocessing of the corpus can further improve the performance of the model A word alignment algorithm. The performance seems to improve after lemmatisation and POS-tagging (POS tagging is a necessary step for morphological lemmatisation), but it seems not profitable to remove the POS tag. Compound splitting seems to be the most rewarding pre-processing step.

Table 7: Number of types and tokens after preprocessing

	raw text	lemma plus POS	lemmatised	lemmatised comp. splitting	noun phrases
English tokens	94118	93985	93985	94118	26297
types	4525	3936	3478	3426	8406
Dutch tokens	108113	107508	107508	113637	29754
types	6897	6459	5704	4443	8665

Noun phrase extraction will drop the performance – especially the recall – of the algorithm. This can be explained by looking at the number of types and tokens of the parallel corpus after the various preprocessing steps. After noun phrase extraction the number of types for English is almost doubled. The number of tokens for English is more than three times less than before. Noun phrase extraction eliminated much of the redundancy in the corpus. This might be an explanation for the drop in performance. All other preprocessing steps decrease the number of types. Compound splitting increases the number of tokens with about 5000 for Dutch[9].

[8] The Netherlands Organisation of Applied Scientific Research.

[9] The number of tokens is not exactly the same for the first three experiments, because the Xerox morphological tools also detect a few multi-word expressions.

7 Future plans

The work presented in this paper is work in progress. The main goal of the Twenty-One project is to build a demonstrator system that supports cross-language information retrieval (CLIR) in the restricted domain of ecology and sustainable development. With the model A version of the word alignment algorithm we introduced a fast and robust algorithm that reliably produces the domain specific translation lexicons needed for CLIR within Twenty-One.

Although model A outperforms model B in this experiment, the possibilities of model B are more promising for further research. It would be interesting to add extra parameters to the model in order to improve its performance. Following the IBM approach (P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer 1993), fertility parameters could be added for both languages instead of adding them just for one language. Another interesting approach would be to include a method for correcting the initial tokenisation of the parallel corpus as proposed by Melamed (I.D. Melamed 1997b) in order to extract the multi-word expressions found by the algorithm explicitly.

Finally we hope to evaluate the performance of the algorithms on bigger and possibly more noisy corpora than the Agenda 21 corpus.

Acknowledgements

I would like to thank the following people: Franciska de Jong for general advise. Wilbert Kallenberg for his help on statistics. David Hull, Eric Gaussier and Anne Schiller of Xerox Research Centre Europe for discussions on word alignment and support on the Xerox morphological tools. Wessel Kraaij and Joost van Surksum of the Netherlands Organisation of Applied Scientific Research (TNO) for general advice and support on the TNO parser. Dan Melamed of the University of Pennsylvania for discussions on bi-directional word alignment models.

References

P.F. Brown, J.C. Cocke, S.A. Della Pietra, V.J. Della Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin (1990), A statistical approach to machine translation., *Computational Linguistics*, 16(2):79–85.

P.F. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer (1993), The mathematics of statistical machine translation: Parameterestimation., *Computational Linguistics*, 19(2):169–176.

A.P. Dempster, N.M. Laird, and D.B. Rubin (1977), Maximum likelihood from incomplete data via the em-algorithm plusdiscussions on the paper., *Journal of the Royal Statistical Society*, 39(B):1–38.

B.S. Everitt, editor (1992), *The analysis of Contingency Tables, second edition.*, Chapman & Hall.

W.A. Gale and K.W. Church (1991), Identifying word correspondences in parallel texts., In *Fourth DARPA Workshop on Speech and Natural Language*, pages152–157.

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

J. Hammersley and D. Handscomb, editors (1964), *Monte Carlo methods.*, Chapman and Hall.

D. Hiemstra (1997a), Deriving a bilingual lexicon for cross language information-retrieval., In *Proceedings of Gronics 1997*, pages 21–26.

D. Hiemstra, F.M.G. de Jong, and W. Kraaij (1997b), Domain specific lexicon acquisition tool for cross-languageinformation retrieval., In *Proceedings of RIAO'97 Conference on Computer-AssistedSearching on the Internet*, pages 255–266.

W. Kraaij (1997b), Multilingual functionality in the TwentyOne project., In *AAAI Symposium on Cross-Language Text and Speech Retrieval*.American Association for Artificial Intelligence, March 1997.

J. Kupiec (1993), An algorithm for finding noun phrase correspondences in bilingualcorpora., In *Proceedings of the 31st Annual Meeting of the Association ofComputational Linguistics*, pages 17–22.

W. Martin and G.A.J. Tops et al., editors (1986), *van Dale Groot woordenboek Nederlands Engels.*, Van Dale Lexicografie bv.

I.D. Melamed (1997b), Automatic discovery of non-compositional compounds in parallel data., In *Second Conference on Empirical Methods in Natural LanguageProcessing (EMNLP'97)*.

I.D. Melamed (1997a), A word-to-word model of translation equivalence., In *Proceedings of the 35th Conference of the Association forComputational Linguistics*.

N.J. Nilsson, editor (1980), *Principles of Artificial Intelligence.*, Tioga Publishing Company.

F. Smadja, K.R. McKeown, and V. Hatzivassiloglou (1996), Translating collocations for bilingual lexicons: A statisticalapproach., *Computational Linguistics*, 22(1):1–38.

Systran (1998), Welcome to systran software, inc., *http://www.systransoft.com/*.

W.G. ter Stal, J-H Beijert, G. de Bruin, J. van Gent, F.M.G. de Jong, W. Kraaij, K. Netter, and G. Smart (1998), Twenty-one: Cross-language disclosure and retrieval of multimediadocuments on sustainable development., *Journal of Computer Networks and ISDN Systems*, to appear.

P. van der Eijk (1993), Automating the acquisition of bilingual terminology., In *Proceedings of the sixth Conference of the European Chapterof the Association for Computational Linguistics*, pages 113–119.

D. Wu and X. Xia (1995), Large-scale automatic extraction of an english-chinese translationlexicon., *Machine Translation*, 9:285–313.