A Simple Hybrid Aligner for Generating Lexical Correspondences in Parallel Texts

Lars AHRENBERG, Mikael ANDERSSON & Magnus MERKEL
NLPLAB, Department of Computer and Information Science
Linköping University, S-581 83 Linköping, Sweden
lah@ida.liu.se, mikander@hotmail.com, magme@ida.liu.se

Abstract

We present an algorithm for bilingual word alignment that extends previous work by treating multi-word candidates on a par with single words, and combining some simple assumptions about the translation process to capture alignments for low frequency words. As most other alignment algorithms it uses co-occurrence statistics as a basis, but differs in the assumptions it makes about the translation process. The algorithm has been implemented in a modular system that allows the user to experiment with different combinations and variants of these assumptions. We give performance results from two evaluations, which compare well with results reported in the literature.

Introduction

In recent years much progress have been made in the area of bilingual alignment for the support of tasks such as machine translation, machine-aided translation, bilingual lexicography and terminology. For instance, Melamed (1997a) reports that his word-to-word model for translational equivalence produced lexicon entries with 99% precision and 46% recall when trained on 13 million words of the Hansard corpus, where recall was measured as the fraction of words from the bitext that were assigned some translation. Using the same model but less data, a French/English software manual of 400,000 words, Resnik and Melamed (1997) reported 94% precision with 30% recall.

While these figures are indeed impressive, more telling figures can only be obtained by measuring the effect of the alignment system on some specific task. Dagan and Church (1994) reports that their Termight system helped double the speed at which terminology lists could be compiled at the AT&T Business Translation Services.

It is also clear that the usability of bilingual concordances would be greatly improved if the system could indicate both items of a translation pair and if phrases could be looked up with the same ease and precision as single words (Macklovitch and Hannan 1996).

For the language pairs that are of particular interest to us, English vs. other Germanic languages, the ability to handle multi-word units adequately is crucial (cf. Jones and Alexa 1997). In English a large number of technical terms are multi-word compounds, while the corresponding terms in other Germanic languages are often single-word compounds. We illustrate with a few examples from an English/Swedish computer manual:

Table 1. Equivalent compounds in an English/Swedish bitext.

English	Swedish
file manager	filhanterare
network server	nätverksserver
operating system	operativsystem
setup directory	installationskatalog

Also, many common adverbials and prepositions are multi-word units, which may or may not be translated as such.

Table 2. Equivalent adverbials and prepositions

English	Swedish
after all	när allt kommer omkring
in spite of	trots
in general	i allmänhet

1. The Problem

The problem we consider is how to find word and phrase alignments for a bitext that is already aligned at the sentence level. Results should be delivered in a form that could easily be checked and corrected by a human user.

Although we primarily use the system for

bitexts with an English and a Scandinavian half, the system should preferably be useful for many different language pairs. Thus we don't rely on the existence of POS-taggers or lemmatizers for the languages involved, but wish to provide mechanisms that a user can easily adapt to new languages.

The organisation of the paper is as follows: In section 2 we relate this approach to previous work, in section 3 we motivate and spell out our assumptions about the behaviour of lexical units in translation, in section 4 we present the basic features of the algorithm, and in section 5 we present results from an evaluation and try to compare these to the results of others.

2. Previous work

Most algorithms for bilingual word alignment to date have been based on the probabilistic translation models first proposed by Brown et al. (1988, 1990), especially Model 1 and Model 2. These models explicitly exclude multi-word units from consideration[1]. Melamed (1997b), however, proposes a method for the recognition of multi-word compounds in bitexts that is based on the predictive value of a translation model. A trial translation model that treat certain multi-word sequences as units is compared with a base translation model that treats the same sequences as multiple single-word units.

A drawback with Melamed's method is that compounds are defined relative to a given translation and not with respect to language-internal criteria. Thus, if the method is used to construct a bilingual concordance, there is a risk that compounds and idioms that translate compositionally will not be found. Moreover, it is computationally expensive and, since it constructs compounds incrementally, adding one word at a time, requires many iterations and much processing to find linguistic units of the proper size.

Kitamura and Matsumoto (1996) present results from aligning multi-word and single word expressions with a recall of 80 per cent if partially correct translations were included. Their method is iterative and is based on the use of the Dice coefficient. Smadja et. al (1996) also use the Dice coefficient as their basis for aligning collocations between English and French. Their evaluation show results of 73 per cent accuracy (precision) on average.

3. Underlying assumptions

As Fung and Church (1994) we wish to estimate the bilingual lexicon directly. Unlike Fung and Church our texts are already aligned at sentence level and the lexicon is viewed, not merely as word associations, but as associations between lexical units of the two languages.

We assume that texts have structure at many different levels. At the most concrete level a text is simply a sequence of characters. At the next level a text is a sequence of word tokens, where word tokens are defined as sequences of alphanumeric character strings that are separated from one another by a finite set of delimiters such as spaces and punctuation marks. While many characters can be used either as word delimiters or as non-delimiters, we prefer to uphold a consistent difference between delimiters and non-delimiters, for the ease of implementation that it allows. At the same time, however, the tokenizer recognizes common abbreviations with internal punctuation marks and regularizes clitics to words (e.g. *can't* is regularized to *can not*).

At the next level up a text can be viewed as a partially ordered bag of lexical units. It is a bag because the same unit can occur several times in a single sentence. It is partially ordered because a lexical unit may extend across other lexical units, as in

He turned the offer down.
Tabs were kept on him.

We say that words express lexical units, and that units are expressed by words. A unit may be expressed by a multi-word sequence, while a given word can express at most one lexical unit.[2]

It is often hard to tell the difference between a lexical unit and a lexical complex. We assume that

[1] Model 3–5 includes multi-word units in one direction.

[2] This latter assumption is actually too strict for Germanic languages where morphological compounding is a productive process, but we make it nevertheless, as we have no means too identify compounds reliably. Moreover, the borderline between a lexicalized compound and a compositional compound is hard to draw consistently, anyway.

recurrent collocations that pass certain structural and contextual tests are candidate expressions for lexical units. If such collocations are found to correspond to something in the other half of the bitext on the basis of co-occurrence measures, they are regarded as expressions of lexical units. This will include compound names such as 'New York', 'Henry Kissinger' and 'World War II', and compound terms such as 'network server directory'. Thus, as with the compositional compounds just discussed, we prefer high recall to high precision in identifying multi-word units.

The expressions of a lexical unit form an equivalence class. An equivalence class for a single-word unit includes its morphological variants. An equivalence class for a multi-word unit should include syntactic variants as well. For instance, the lexical unit *turn_down* should include 'turned down' 'turning down' as well as expressions where the particle is separated from the verb by some appropriate phrase, as in the example above. The current system, though, does not provide for syntactic variants.

Our aim is to establish relations not only between corresponding words and word sequences in the bitext, but also between corresponding lexical units. A problem is then that the algorithm cannot recognize lexical units directly, but only their expressions. It helps to include lexical units in the underlying model, however, as they have explanatory value. Moreover, the algorithm can be made to deliver its output in the form of correspondences between equivalence classes of expressions belonging to the same lexical unit.

For the purpose of generating the alignment and the dictionary we divide the lexical units into three classes:

1. irrelevant units,
2. closed class units,
3. open class units

The same categories apply to expressions.

Irrelevant units are simply those that we don't want to include. They have to be listed explicitly. The reason for not including some items may vary with the purpose of alignment. Even if we wish the alignment to be as complete as possible, it might be useful to exclude certain units that we suspect may confuse the algorithm. For instance, the do-support found in English usually has no counterpart in other languages. Thus, the different forms of 'do' may be excluded from consideration from the start.

As for the translation relation we make the following assumptions:

1. A lexical unit in one half of the bitext corresponds to at most one lexical unit in the other half. This can be seen as a generalization of the one-to-one assumption for word-to-word translation used by Melamed (1997a) and is exploited for the same purpose, i.e. to exclude large numbers of candidate alignments, when good initial alignments have been found.

2. Open class and closed class lexical units are usually translated and there are a limited number of lexical units in the other language that are commonly used to translate them. While deliberately vague this assumption is what motivates our search for frequent pairs <source expression, target expression> with high mutual information. It also motivates our choice of regarding additions and deletions of lexical units in translation as haphazard apart from the case of a restricted set of irrelevant units that we assume can be known in advance.

3. Open class units can only be aligned with open class units, and closed class units can only be aligned with closed class units. This assumption seems generally correct and has the effect of reducing the number of candidate alignments significantly. Closed class units have to be listed explicitly. The assumption is that we know the two languages sufficiently well to be able to come up with an appropriate list of closed class units and expressions. Multi-word closed class units are listed separately. Closed class units can be further classified for the purposes of alignment (see below).

4. If some expression for the lexical unit U_S is found corresponding to some expression for the lexical unit U_T, then assume that any expression of U_S may correspond to any expression of U_T. This assumption is in accordance with the often made observation that morphological properties are not invariants in translation. It is used to make the algorithm more greedy by accepting infrequent alignments that are morphological variants of high-rating ones.

5. If one half of an aligned sentence pair is the expression of a single lexical unit, then assume that the other half is also. This is definitely a heuristic, but it has been shown to be very useful

for technical texts involving English and Scandinavian, where terms are often found in lists or table cells (Tiedemann 1997). This heuristic is useful for finding alignments regardless of frequencies.

Similarly, if there is only one non-aligned (relevant open class) word left in a partially aligned sentence, assume that it corresponds to the remaining (relevant open class) words of the corresponding sentence.

6. Position matters, i.e. while word order is not an invariant of translation it is not random either. We implement the contribution of position as a distribution of weights over the candidate pairs of expressions drawn from a given pair of sentences. Expressions that are close in relative position receive higher weights, while expressions that are far from each other receive lower weights.

4. The Approach

4.1 Input

A bitext aligned at the sentence level.

4.2 Output

There are two types of output data: (i) a table of link types in the form of a bilingual dictionary where each entry has the form $<<s,t^1,...t^n>>$, s being the source expression type and $t^1,...t^n$ the target expression types that were found to correspond to s; and (ii) a table of link instances $<<s,t><i,j>>$ sorted by sentence pairs, where s is some expression from the source text, t is an expression from the translated text, and i and j are the (within-sentence) positions of the first word of s and t, respectively.

4.3 Preprocessing

Both halves of the bitext are regularized.

When open class multi-word units are to be included, they are generated in a preprocessing stage for both the source and target texts and assembled in a table. For this purpose, we use the phrase extracting program described in Merkel et al. (1994).

4.4 Basic operation

The basic algorithm combines the K-vec approach, described by Fung and Church (1993), with the greedy word-to-word algorithm of Melamed (1997a). In addition, open class expressions are handled separately from closed class expressions, and sentences consisting of a single expression are handled in the manner of Tiedemann (1997).

The algorithm is iterative, repeating the same process of generating translation pairs from the bitext, and then reducing the bitext by removing the pairs that have been found before the next iteration starts. The algorithm will stop when no more pairs can be generated, or when a given number of iterations have been completed.

In each iteration, the following operations are performed:

(i) For each open class expression in the source half of the bitext (with frequency higher than 3), the open class expressions in corresponding sentences of the other half are ranked according to their likelihood as translations of the given source expression.

We estimate the probability that a candidate target expression is a translation by counting co-occurrences of the expressions within sentence pairs and overall occurrences in the bitext as a whole. Then the t-score, used by Fung and Church, is calculated, and the candidates are ranked on the basis of this value:

In our case K is the number of sentence pairs in

$$t \approx \frac{prob(V_s,V_t) - prob(V_s)prob(V_t)}{\sqrt{\frac{1}{K}prob(V_s,V_t)}}$$

the bitext. The target expression giving the highest t-score is selected as a translation provided the following two conditions are met: (a) this t-score is higher than a given threshold, and (b) the overall frequency of the pair is sufficiently high. (These are the same conditions that are used by Fung and Church.)

This operation yields a list of translation pairs involving open class expressions.

(ii) The same as in (i) but this time with the closed class expressions. A difference from the previous stage is that only target candidates of the proper sub-category or sub-categories for the source expression are considered. Conjunctions and personal pronouns are for example specified for both the target and the source languages. This strategy helps to limit the search space when closed-class expressions are linked.

(iii) Open class expressions that constitute a sentence on their own (not counting irrelevant word tokens) generate translation pairs with the open class expressions of the corresponding sentence.

(iv) When all (relevant) source expressions have been tried in this manner, a number of translation pairs have been obtained that are entered in the output table and then removed from the bitext. This will affect t-scores by reducing mariginal frequencies and will also cause fewer candidate pairs to be considered in the sequel. The reduced bitext is input for the next iteration.

4.5 Variants

The basic algorithm is enhanced by a number of modules that can be combined freely by the user. These modules are

- a *morphological module* that groups expressions that are identical modulo specified sets of suffices;
- a *weight module* that affects the likelihood of a candidate translation according to its position in the sentence;
- a *phrase module* that includes multi-word expressions generated in the pre-processing stage as candidate expressions for alignment.

4.5.1 The morphological module

The morphological module collects open class translation pairs that are similar to the ones that are found by the basic algorithm. More precisely, if the pair (X, Y) has been generated as a translation pair in some iteration, other candidate pairs with X as the first element are searched. A pair (X, Z) is considered to be a translation pair iff there exist strings W, F and G such that

$$Y = WF,$$
$$Z = WG$$

and F and G have been defined as different suffices of the same paradigm.

The data needed for this module consists of simple suffix lists for regular paradigms of the languages involved. For example, *[0, s, ed, ing]* is a suffix list for regular English verbs. They have to be defined by the user in advance.

When the morphological module is used, it is possible to *reverse* the direction of the linking process at a certain stage. After each iteration of linking expressions from source to target, the different inflectional variants of the target word are used as input data and these candidates are then linked from target to source. This strategy makes it possible to link low-frequency source expressions belonging to the same suffix paradigm.

4.5.2 The weight module

The weight module distribute weights over the target expressions depending on their position relative to the given source expression. The weights must be provided by the user in the form of lists of numbers (greater than or equal to 0).

The weight for a pair is caclulated as the sum of the weights for the instances of that pair. This weight is then used to adjust the co-occurrence probabilities by using the weight instead of the co-occurrence frequency as input to the the t-score formula. The threshold used is adjusted accordingly. In the current configuration of weights, the threshold is increased by 1. In the weight module it is possible to specify the maximal distance between a source and target expression measured as their relative position in the sentences.

4.5.3 The phrase module

When the phrase module is invoked, multi-word expressions are also considered as potential elements of translation pairs. The multi-word expressions to be considered are generated in a special pre-processing phase and stored in a phrase table.

T-scores for candidate translation pairs involving multi-word expressions are calculated in the same way as for single words. When weights are used the weight of a multi-word expression is considered equal to that of its first word.

It can happen that the t-scores for two pairs $<s,t^1>$ and $<s,t^2>$, where t^1 is a multi-word expression and t^2 is a word that is part of t^1, will be identical or almost identical. In this case we prefer the almost identical target multi-word expression over a single word candidate if it has a t-value over the threshold and is one of the top six target candidates. When a multi-word expression is found to be an element of a translation pair, the expressions that overlap with it, whether multi-word or single-word expressions, are removed from the current agenda and not considered until the next iteration.

5. Evaluation

The algorithm was tested on two different texts; one novel (66,693 source words) and one computer program manual (169.779 source words) which both were translated from English into Swedish. The tests were run on a Sun UltraSparc1 Workstation with 320 MB RAM and took 55 minutes for the novel and 4 and a half hour for the program manual.

The tests were run with three different configurations on each text: (i) the baseline (*B*) configuration which is the t-score measure, (ii) all modules except the weights module (*AM-W*), but a linkdistance constraint was used and set to 10; and (iii) all modules (*AM*) including morphology, weights and phrases. The t-score threshold used was 1.65 for B and AM-W, and 2.7 for AM, the minimum frequency of source expression was set to 3. Closed-class expressions were linked in all configurations. In the baseline configuration no distinction was made between closed-class and open-class expressions. In the AM-W and AM tests the closed-class expressions were divided into different subcategories and at the end of each iteration the linking direction was reversed at the end of each of the six iterations which improves the chances of linking low frequency source expressions. The characteristics of the source texts used are shown in Table 3.

Table 3. Characteristics for the two source texts

	Novel	Prog. Man.
Size in running words	66,693	169,779
No of word types	9,917	3,828
Word types frequency 3 or higher	2,870	2,274
Word types frequency 2 or 1	7,047	1,554
Multi-word expression types (found in pre-processing)	243	981

The novel contains a high number of low frequency words whereas the program manual contains a higher proportion of words that the algorithm acturally tested as the frequency threshold was set to 3.

The results from the tests are shown in Table 4. The evaluation was done on an extract from the automatically produced dictionary. All expressions starting with the letters N, O and P were evaluated for all three configurations of each text.

The results from the novel show that recall is almost tripled in the sample, from 234 in the B configuration to 709 linked source expressions with the AM configuration. Precision values for the novel lie in the range from 90.13 to 92.50 per cent when partial links are judged as errors and slightly higher if they are not. The use of weights seems to make precision somewhat lower for the novel which perhaps could be explained by the fact that the novel is a much more varied text type.

For the program manual the recall results are as good as for the novel (three times as many linked source types for the AM configuration compared to baseline). Precision is increased, but perhaps not

Table 4. Results from two bitexts, using T-score only (B), all modules except the weights (AM-W), and all modules (AM)

	Novel			Program Manual		
	B	AM-W	AM	B	AM-W	AM
Linked source expressiones	1,575	2,467	2,895	1,631	2,748	2,878
Linked multi-word expr.	0	177	187	0	683	734
Link types in total	2,059	4,833	5,754	2,740	7,241	7,487
Links in evaluated sample	234	573	709	318	953	1,005
Correct links in sample	207	530	639	199	655	753
Errors in sample	21	19	30	51	137	122
Partial links in sample	6	24	40	68	161	130
Precision	88.46%	92.50%	90.13%	62.58%	68.73%	74.93%
Precision (only errors)	91.03%	96.68%	95.77%	83.96%	85.62%	87.86%
Token recall	50.9%	54.6%	56.70%	60.2%	67.1%	67.3%
Type recall freq 3 or higher	54.88%	72.06%	82.65%	73.88%	82.10%	85.53%
Type recall freq 2 or 1	0	3.15%	4.87%	0	12.74%	12.74%

to the level we anticipated at first. Multi-word expressions are linked with a relatively high recall (above 70%), but the precision of these links are not as high as for single words. Our evaluations of the links show that one major problem lies in the quality of the multi-word expressions that are fed into the alignment program. As the program works iteratively and in the current version starts with the multi-word expressions, any errors at this stage will have consequences in later iterations.

We have run each module separately and observed that the addition of each module improves the baseline configuration by itself. To compare our results to those from other approaches is difficult. Not only are we dealing with different language pairs but also with different texts and text types. There is also the issue of different evaluation criteria. A pure word-to-word alignment cannot be compared to an approach where lexical units (both single word expressions and multi-word expressions) are linked. Neither can the combined approach be compared to a pure phrase alignment program because the aims of the alignment are different.

However, as far as we can judge given these difficulties, the results presented in this paper are on par with previous work for precision and possibly an improvement on recall because of how we handle low-frequency variants in the morphology module and by using the single-word-line strategy. The handling of closed-class expressions have also been improved due to the division of these expressions into subcategories which limits the search space considerably.

Acknowledgements

This work is part of the project "Parallell corpora in Linköping, Uppsala and Göteborg" (PLUG), jointly funded by Nutek and HSFR under the Swedish National research programme in Language Technology.

References

Brown, P.F., J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, & P. Roossin. (1988) "A Statistical Approach to Language Translation." *Proceedings of the 12th International Conference on Computational Linguistics*. Budapest.

Brown, P F, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, R. Mercer, & P. Roossin. (1990) "A Statistical Approach to Machine Translation." *Computational Linguistics 16(2).*

Dagan, I, & K. W. Church. (1994) "Termight: Identifying and Translating Technical Terminology." *Proceedings from the Conference on Applied Natural Language Processing;* Stuttgart.

Fung, P, & K. W. Church. (1994) "K-vec: A New Approach for Aligning Parallel Texts." *Proceedings from the 15th International Conference on Computational Linguistics*, Kyoto.

Jones, D: & M. Alexa (1997) "Towards automatically aligning German compounds with English word groups." In *New Methods in Language Processing* (eds. Jones D. & H. Somers). UCL Press, London.

Kitamura, M. & Y. Matsumoto (1996) "Automatic Extraction of Word Sequence Correspondences in Parallel Corpora". In Proceedings of the Fourth Annual Workshop on Very Large Corpora (WVLC-4), Copenhagen.

Macklovitch, E., & Marie-Loiuse Hannan. (1996) "Line 'Em Up: Advances in Alignment Technology and Their Impact on Translation Support Tools." In *Proceedings of the Second Conference of the Association for Machine Translation in the Americas*, Montreal.

Melamed, I. D. (1997a) "A Word-to-Word Model of Translational Equivalence." *Proceedings of the 35th Conference of the Association for Computational Linguistics*, Madrid.

Melamed, I. Dan. (1997b) "Automatic Discovery of Non-Compositional Compounds in Parallel Data." Paper presented at the 2nd Conference on Empirical Methods in Natural Language Processing, Providence.

Merkel, M. B. Nilsson, & L. Ahrenberg, (1994) "A Phrase-Retrieval System Based on Recurrence." In Proceedings of the Second Annual Workshop on Very Large Corpora (WVLC-2). Kyoto.

Resnik, P. & I. D. Melamed. (1997) "Semi-Automatic Acquisition of Domain-Specific Translation Lexicons." In *Proceedings of the 7th ACL Conference on Applied Natural Language Processing*. Washington DC.

Smadja F., K. McKeown, & V. Hatzivassiloglou, (1996) "Translating Collocations for Bilingual Lexicons: A Statistical Approach." In *Computational Linguistics, Vol. 22 No. 1.*

Tiedemann, Jörg. (1997) "Automatic Lexicon Extraction from Aligned Bilingual Corpora." Diploma Thesis, Otto-von-Guericke-Universität Magdeburg.