The Use of Approximate String Matching Techniques in the Alignment of Sentences in Parallel Corpora

Anthony M. McEnery, Michael P. Oakes & R. G. Garside

The University of Lancaster, UK

Abstract

Parallel corpora such as the Canadian Hansard corpus and the International Telecommunications Union (ITU) corpus each provide the same text in two or more languages, and have been aptly described as the "Rosetta Stone" of modern corpus linguistics [1]. Their use within MT is burgeoning, permeating all levels of the discipline, and even being used as the basis of full-blown statistically based MT systems.

This paper will concern itself with the task of automatic bilingual lexicon construction, which is one of the major goals of the CRATER project ("Corpus Resources and Terminology Extraction", funded under the MLAP initiative of the CEC, grant number MLAP-93/20).

The approach to bilingual lexicon alignment taken here entails the alignment of corpora, and then a detailed search through the corpus for lexical cognates. Consequently the paper will begin with a brief discussion of the alignment procedures used on the project to date, and move to a discussion of various similarity metrics used to evaluate lexical similarity.

1 Introduction: Language independent and language-pair specific approaches to alignment

Parallel Corpora provide an ideal test-bed for many tasks, such as translation tuition and the production of probabilistic dictionaries. To be of use, however, they must first be aligned, so that it is known which segments in one corpus correspond with which segments in the other. Current research on the CRATER project concerns automatic alignment of parallel texts both at the sentence and word level.

Two purely statistical approaches to parallel corpus alignment are the Gale & Church [2] and Kay & Röscheisen [3] statistical alignment algorithms. The Gale and Church approach is determined by two factors, namely:

1. The relative lengths in characters of the sentences to be aligned

2. The *a-priori* likelihood of the type of alignment, where for example a simple substitution (a single sentence of one language being translated by a single sentence of the other language) is more likely than an "expansion" (a single sentence of one language being translated by two sentences of the other language).

We have re-implemented this algorithm based on the code given in [2], and have obtained success rates of 98% for 100 sentences of the English/French section of the ITU corpus (which is a literal English to French translation), 75% for news items in English and German, and between 69.5% and 100% for short passages in English and Polish.

The Kay & Röscheisen algorithm differs quite radically in its approach from the Gale & Church model. It first makes a rough estimate of which sentences could be aligned, then uses this information to find which words within those sentences are most likely to be translations of each other. This data is then used to produce improved alignment at the sentence level, leading in turn to the discovery of further correspondences at the word level. This iterative process continues either for a set number of iterations or until no further updates occur.

Both approaches are theoretically language independent, but we feel that the accuracy and speed of these algorithms could be improved by the incorporation of language-pair specific information, in the form of cognates. Cognates are pairs of words which are reliable translations of each other, and provide “anchor” points for text alignment in bilingual corpora, where an anchor point is a point of known correspondence between the two texts.

Approximate string matching techniques have traditionally been employed in information retrieval for the recognition of different grammatical forms of the same root word or lemma. In this paper we extend their use to the identification of word clusters in different languages which are related in meaning. We employ the resulting information to assist in the task of identifying language-pair specific cognates for sentence alignment in our multilingual corpora. The approximate string-matching techniques we have examined to date are Dice's Similarity Coefficient, Truncation and Dynamic programming [5]. These techniques are language specific in that a greater number of true cognates will be discovered for more closely related language pairs.

2 Dice's Similarity Coefficient.

Dice's Similarity Coefficient [6] is a metric originally developed for the comparison of biological specimens, which returns a real numeric value in the range 0 to 1. This metric can be employed to assign a value to describe the lexical similarity between two words, in this case one word from the English token or lemma list, and one from the corresponding French token or lemma list produced from the International Telecommunications Union (ITU) corpus. Two terms which are totally dissimilar in their character structures would return a Dice score of 0, while two lexically identical terms would return a Dice Score of 1. Using the technique of Adamson & Boreham [7], the comparison of words depends on the number of matching bigrams or consecutive pairs of characters. The formula for Dice's Similarity Coefficient, S, is then

$$S = 2a / (b + c)$$

where a is the number of matching bigrams and b and c are the total number of bigrams in each term.

For example, the English word “colour” yields the bigrams “co-ol-lo-ou-ur”, while the French word “couleur” yields the bigrams “co-ou-ul-le-eu-ur”. The total number of matching bigrams is 3, while the total number of bigrams is 11. The coefficient of similarity between these terms is then 6/11, about 0.55.

An experiment was performed in which Dice's Similarity Coefficient was found for each term pair in the English and French ITU corpus token lists, which consist of one occurrence of every single lexical item in the corpus. The term pairs were divided into bands according to their respective Dice scores, each band having a bandwidth of 0.1. In Tables 5 - 1 and 5 - 2, the lower limits of the bands are given. All term pairs with a Dice score of less than 0.4 were disregarded. A pseudo-random sample of 200 pairs from each band was printed out, and the pairs were each scored : 1 if they were considered either exact translations or useful grammatical variants of each other (i.e. a sentence containing one of the pair might conceivably contain the other in its translation), or 0 if the terms were either unrelated, matching only in their prefixes and suffixes, or terms where an identical root had given rise to variants of considerably different meaning (e.g. “recover”,

“decouvrir”). The proportion of pairs with a score of 1 in each band is given in the row labelled "tokens" in Table 5 - 1.

The experiment was repeated using the corresponding ITU lemma lists, which differ from the token lists in that plurals and inflexional variants of a word are all considered as a single word rather than separate items. The results are shown in the row labelled "lemmas" in Table 5 - 1.

Dice =	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Tokens	0	.07	.205	.475	.89	.98	1
Lemmas	.005	.02	.175	.535	.525*	.965	1

Table 5 - 1 Success rate for determining cognates by Dice's Similarity Coefficient: All pairs considered.

In most cases, better results were obtained for the token lists than for the lemma lists. This was because the lemmatisation process results in the removal of suffixes which often correspond in true cognates in English and French, since the suffix "s" denotes plurality in both languages. The asterisk * denotes that unexpectedly poor results were obtained in the 0.8 band for the lemmas, due to a repeated phenomenon, namely the comparison of a four character acronym in one language with a three character acronym in the other, the two acronyms in question not being translations of each other. In order to circumvent this problem, the experiments were repeated considering only those term pairs in which both members each consisted of at least 4 characters. The results for this evaluation are given in Table 5 - 2.

Dice =	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Tokens	0	.07	.215	.49	.816	.97	1
Lemmas	0	.05	.225	.48	.71	.95	1

Table 5 - 2 Success rate for determining cognates by Dice's Similarity Coefficient: Pairs in which each term consists of 4 or more characters.

From the results given in Tables 5 - 1 and 5 - 2, it was decided that Dice's Similarity Coefficient should be used in conjunction with the token list rather than the lemma list for cognate matching, and only terms with 4 or more characters should be considered. When matching pairs across the entire corpus, a Dice score of at least 0.9 is required to ensure 95 % accuracy for cognate identification. As shown in Table 5 - 3, the estimated number of true cognates in the corpus (found by multiplying the number of word pairs retrieved in each band by the percentage correct in that band) is greatest in the lower scoring bands. However, as will be expanded upon later, lower Dice scores can identify cognates accurately when only those word pairs which occur in corresponding regions of text are considered.

Score	Number in Band	% Correct	Estimated Number of Cognates
1.0	29259	100	29259
0.9	20142	97.0	19538
0.8	57150	81.6	46634
0.7	118080	49.0	57859
0.6	372528	21.5	80094
0.5	1547289	7.0	108310
0.4	5508639	0.0	0

Table 5 - 3 Potential numbers of cognates retrievable from each Dice band.

3 Truncation

Term truncation is commonly employed in the field of Information Retrieval to enable searching for clusters of related terms as a single unit. The technique is simply to regard two terms which commence with the same *n* characters as equivalent. An experiment was performed in which each term in the English token list was compared with each term in the French token list, and the number of initial common characters in each case was recorded. All term pairs with fewer than 3 initial characters in common were disregarded. All other pairs were divided into bands according to their number of common initial characters, and a pseudo-random sample of 200 pairs was then taken from each band for evaluation. As before, a score of 1 for each pair was assigned if the two terms were considered useful grammatical variants of each other, and a score of 0 otherwise. The proportion of pairs which were considered true cognates is given in Table 5 - 4.

Length	3	4	5	6	7	8	9	>9
Correct	.01	.075	.14	.685	.925	.975	.995	1

Table 5 - 4 Accuracy of cognate determination using truncation data without stoplisting.

The results in Table 5 - 4 show that cognates can be found with 95 % reliability if both terms commence with the same 8 characters.

As shown in Table 5 - 5, an estimation of the number of true cognates in the corpus which could potentially be found at each truncation level was made by multiplying the number of matching pairs for each truncation length by the percentage of correct pairs found in each sample of 200.

The resulting values show that the bulk of true cognates are to be found using truncation lengths in the range 5 to 7. Thus it was necessary to devise some means of rendering the truncation matching process more accurate in this range. The estimated number of true cognates retrieved by truncation at the 95% accuracy level (sum of the estimated numbers of true cognates retrieved by trancation to 8 characters or more) was 13,328 compared with the corresponding value for Dice's similarity coefficient (derived from Table 5 - 3, with scores ≥ 0.9) of 48,797. We feel that both techniques should be used in conjunction, to maximise the number of retrieved cognates.

Truncation Length	Total Matches	% Correct	Estimated number of Cognates
20	3	100	3
19	9	100	9
18	14	100	14
17	19	100	19
16	16	100	16
15	71	100	71
14	175	100	175
13	231	100	231
12	539	100	539
11	891	100	891
10	1987	100	1987
9	3516	99.5	3498
8	6025	97.5	5874
7	10591	92.5	9797
6	17685	68.5	12114
5	71875	14	10063
4	112423	7.5	8432
3	506420	1	5064

Table 5 - 5 Potential numbers of cognates retrievable at each truncation level.

The observed inaccuracy at shorter truncation lengths resulted from overlap of initial character sequences in unrelated word families, such as in the case of "computer" and "compulsion". To overcome this problem, stoplists were created for each truncation length. In the example given here, a rule should be created to the effect that "if both terms truncate to the same 5 characters, and these characters are 'compu', then the word pair should not be considered a cognate". The procedure for creating the stoplists for each truncation length was as follows:

a) 1000 examples of word pairs with the same initial *n* characters were examined manually.

b) All initial character sequences for incorrect pairs were placed in the stoplist.

c) All initial character sequences which produced at least 10 correct pairs and no incorrect pairs were placed in a "go" list.

d) Return to (a) employing the current stop and go lists.

e) Stop when all pairs allowed through by the stop and go lists have been examined.

Since fully comprehensive stoplists would be unduly large, it was decided to restrict the stoplists to the 100 most commonly employed entries, as determined by a version of the truncation program which kept a record of the number of times each stoplist entry was employed. All stoplist items of length 7 or more were retained, as were items of length 6 firing 4 times or more, length 5 firing 42 times or more, length 4 firing 120 times or more and length 3 firing 360 times or more. The lists of most commonly employed stoplist items for each truncation length are available from the authors.

To estimate the accuracy of truncation after stoplisting, a pseudo-random sample of 200 pairs from each truncation length band was taken. The results are given in Table 5 - 6. The accuracy of truncation matching in the range 5 to 7 is somewhat improved relative to the values given in Table 5 - 4. Accuracy at 5 and 6 might be increased to the 95 % level by the inclusion of further stoplist items.

Length	3	4	5	6	7	8	9	10	11
Correct	4.5%	19.5%	73.5%	93.5%	97.5%	99.5%	99.0%	99.5%	100.0%

Table 5 - 6 Accuracy of truncation after stoplisting.

One possibility originally considered was that the truncation matching procedure might depend solely on the presence of the matching sequence in the go lists, and thus an estimation was made of the retrieval potential and accuracy of truncation using only items in the go lists firing 10 or more times.

In this experiment, only term pairs which commenced with character sequences given in the go lists were considered. A sample of over 200 output pairs was examined for each truncation length, and the results are given in Table 5 - 7. The go lists employed are available on request from the authors. It was found that the use of go lists was over 95% accurate for truncation lengths of 4 or more, but this method yielded relatively few cognates. The danger of go lists, even if found to be sufficiently accurate in one domain, is that counter examples might be found in another domain. With stoplisting, the knowledge that confusion of certain word families is possible will be applicable in all domains.

Length	Number of Rules	Total Matches	% True Cognates	Estimated Number of Cognates
9	58	1827	96.2	1757
8	72	2348	98.3	2308
7	80	3390	98.5	3339
6	73	3571	98.4	3514
5	72	3368	97.7	3291
4	27	1211	95.4	1155
3	13	317	77.6	246

Table 5 - 7 Effectiveness of "Go" lists.

4 Dynamic programming.

The Damerau-Levenshtein metric was originally developed to examine the problem of misspellings in words. It is a procedure for checking whether two character strings differ in any of the following respects, which correspond to the most common errors: single character omissions, insertions, substitutions and reversals. By using sequences of such operations any character string can be transformed into any other character string. Dynamic programming enables one to determine the least number of such transformations required to convert one string into another, and thus provides a measure of the lexical similarity of two strings [5].

We employed the dynamic programming approach to measure the similarity between each English term and each French term in the ITU token lists. In order to convert the integral score produced by this method into a real valued metric in the range 0 to 1, we employed the following formula:

DP Score = Minimum number of substitutions required / Number of characters in the longer term.

In order to evaluate the effectiveness of the Dynamic Programming technique, we found the DP Score for each token pair in the two language lists, and then determined manually the percentage of true cognates in a pseudo-random sample of 250 term pairs in each score band with bandwidth 0.1 in the range 0.4 to 1.0. The results are given in Table 5 - 8. Although it was expected that dynamic programming would produce results similar to Dice's similarity coefficient both in retrieval effectiveness and in terms of the actual token pairs retrieved, we discovered that Dynamic Programming performed less well than Dice's similarity coefficient. The advantage of using Dynamic Programming, however, is that we will be able to maintain counts of which character substitutions regularly occur when examining related terms in two languages.

DP Score	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	.008	.016	.124	.388	.728	.948	1

Table 5 - 8 Accuracy of Dynamic Programming.

5 Repeat of Dice and truncation experiments for restricted regions of text.

The results obtained in Tables 5 - 1 to 5 - 7 were obtained when matching pairs from all regions of the corpus were considered. However, it was expected that when matching pairs in which each member originated from a corresponding region of text such as an equivalent paragraph or sentence, a higher proportion of matching pairs would be true cognates. Thus the experiments were repeated, considering only those pairs found in matching regions of the text.

The two levels of granularity employed were the "hard" and "soft" regions defined by Gale & Church [2]. Hard regions are those regions, often corresponding to paragraphs, within which their alignment algorithm is constrained. No sentence may be aligned with a region of text unless it occurs within the corresponding hard region. Soft regions are the smallest unbreakable regions of text considered by the alignment algorithm. For the purposes of these experiments, the hard and soft regions in the test corpus were determined manually. The programs for these evaluations were produced by Eric Gaussier at IBM Paris, and the results are given in Tables 5 - 9 and 5 - 10.

a) Comparison within Hard Regions, All matches considered:

Dice =	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	.049	.334	.660	.942	.961	.992	.993

b) Comparison within Soft Regions, All matches considered:

Dice =	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	.146	.693	.706	.971	1.00	1.00	1.00

c) Comparison within Hard Regions, Best Match Criterion employed:

Dice =	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	.260	.715	.805	1.00	1.00	1.00	1.00

d) Comparison within Soft Regions, Best Match Criterion employed:

Dice =	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Accuracy	.411	.927	.902	1.00	1.00	1.00	1.00

Table 5 - 9 Dice's Similarity Coefficient

The "Best Match" criterion of Gaussier et al [8] ensures that an English word can only be assigned to a French word if the French word is not associated with another English word with a higher score. In this way, only the term pair with the highest matching score is retained out of all term-pairs with a matching coefficient above threshold.

The results given in Tables 5 - 9 and 5 - 10 show that for Truncation and Dice's Similarity Coefficient, the narrower the region in which term pair comparison is constrained, the more likely it is that two terms with a given score of association are indeed cognates. A similar effect is produced by employing the best match criterion of Gaussier et al. In the case of Truncation, the results could be improved yet further by making use of the stoplist. When the experiments were repeated for Truncation for comparisons within soft regions, no false matches were produced in the samples at any truncation length, irrespective of whether hard or soft regions or the best match criterion were employed.

With regard to sample size, the entries in Table 5 - 9 were obtained with an average of 194 token pairs per cell, the entries in Table 5 - 10 were obtained with an average of 162 token pairs per cell.

As shown in Tables 5 - 1 and 5 - 4, when comparisons are made across the entire corpus, with no Best Match criterion or stoplist, to obtain 95% accuracy requires that thresholds of 0.9 for Dice and 8 for Truncation length must be employed. However, under optimal conditions, this level of accuracy can be obtained for Dice Scores of 0.7 and Truncation lengths of 3. If these measures were employed under these conditions for the entire corpus, the estimated number of true cognates retrieved, derived from Tables 5 - 3 and 5 - 5, would be about 153,000 for Dice and 59,033 for Truncation, these values being the sums of the estimated number of cognates in acceptable bands.

a) Comparison within Hard Regions, All matches considered, No Stoplist employed:

Length	3	4	5	6	≥7
Accuracy	.315	.829	.622	.979	1.00

b) Comparison within Hard Regions, All matches considered, Stoplist employed:

Length	3	4	5	6	≥7
Accuracy	.848	.987	1.00	.992	1.00

c) Comparison within Hard Regions, Best match criterion, No Stoplist employed:

Length	3	4	5	6	≥7
Accuracy	.622	.960	.955	1.00	1.00

d) Comparison within Soft Regions, Best match criterion, Stoplist employed:

Length	3	4	5	6	≥7
Accuracy	.886	1.00	1.00	1.00	1.00

Table 5 - 10 Truncation

6 Future Work: The incorporation of cognate data into statistical alignment algorithms.

We are currently working on the production of a program which will incorporate cognate data into the Gale & Church algorithm. The current algorithm depends on two statistical criteria, namely the relative lengths of the two alignable regions and the likelihood of alignment type relative to a simple 1:1 substitution. We propose that a third factor be included into the calculation, being a measure of the degree of overlap between the sets of cognate terms contained within the alignable regions.

The procedure will be first to produce token lists for each language within a given hard region. Either Dice or Truncation, using the Best Match procedure, will be employed to create a table of term pairs with an associated probability of true cognateness, obtained using an experimentally derived baseline curve derived from the values given in Tables 5 - 9(c) and 5 - 10(c). As dynamic programming considers each putative alignment, a variant of Dice's similarity coefficient designed to deal with real values will again be used, this time to consider the degree of match between the number of cognates present in each segment of text.

For example, consider a putative alignment in which two sentences of English are to be aligned with one sentence of French. Suppose that English sentence 1 contains the terms *a* and *b*, English sentence 2 contains *c* and *d*, while the French sentence contains the terms *a′, b′* and *c′*. According to the cognate table, *a* and *a′*, *b* and *b′*, *c* and *c′* are all cognate pairs with a probability of 0.9, while *d* and *d′* (absent from the French sentence) are a cognate pair with a probability of 0.5. Dice's Similarity Coefficient between the two alignable regions is then

$$2 * (0.9 + 0.9 + 0.9) / (0.9 + 0.9 + 0.9 + 0.9 + 0.9 + 0.9 + 0.5)$$

i.e. the sum of the probability scores for each individual term involved in a match divided by the sum of the probabilities of each individual term which occurs in the putative alignment, resulting in a score of .915.

Incorporation of cognate data directly into the Kay & Röscheisen algorithm should be done at the level of the Word Alignment Table 5 - (WAT). The Dice and Truncation programs will output data in the required format, namely "English_Word, French_Word, Score", and the existing criteria given by Kay & Röscheisen regarding the minimum score and the number of occurrences required to warrant the inclusion of a word pair in the WAT can be employed.

7 Conclusions

We have employed approximate string matching techniques to demonstrate empirically that the higher the degree of lexical match between two words, one taken from each of two languages, the greater the probability that these two words are translations of each other. Word pairs thus deemed to be likely mutual translations can be regarded as cognates, and we have described methods of employing cognate information to enhance the performance of both the Gale & Church and Kay & Röscheisen statistical sentence alignment algorithms.

References

[1] McEnery, A. and Wilson, A. (1993). "Corpora and Translation", UCREL Technical Papers Series Number 2, Department of Linguistics, Lancaster University.

[2] Gale, W.A., & Church, K.W., (1993). "A Program for Aligning Sentences in Bilingual Corpora", Computational Linguistics 19:1, 75-102.

[3] Kay, M., & Röscheisen M. (1993). "Text-Translation Alignment", Computational Linguistics, 19:1, 121-142.

[4] Simard, M., Foster, G. & Isabelle, P. (1992). "Using Cognates to Align Sentences in Bilingual Corpora". Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI92), Montreal, Canada. 67-81.

[5] Hall , P. A. V., & Dowling , G. R. (1980). "Approximate String Matching", Computing Surveys, 12:4, 381-402.

[6] Dice, L. R. (1945) “Measures of the Amount of Ecologic Association Between Species”, Ecology 26, 297-302.

[7] Adamson, G. W. & Boreham, J. (1974). “The Use of an Association Measure Based on Character Structure to Identify Semantically Related Pairs of Words and Document Titles”, Information Storage & Retrieval 10, 253-260.

[8] Gaussier, E., Langé, J.-M., & Meunier, F. (1992) "Towards Bilingual Terminology", 19th International Conference for Literary and Linguistic Computing (ALLC-ACH 1992), Oxford University Press, 121-124.