Acta Cybernetica **18** (2008) 463–478.

Sentence Alignment of Hungarian-English Parallel Corpora Using a Hybrid Algorithm

Krisztina Tóth, Richárd Farkas, and András Kocsor*

Abstract

We present an efficient hybrid method for aligning sentences with their translations in a parallel bilingual corpus. The new algorithm is composed of a length-based and anchor matching method that uses Named Entity recognition. This algorithm combines the speed of length-based models with the accuracy of anchor finding methods. The accuracy of finding cognates for Hungarian-English language pair is extremely low, hence we thought of using a novel approach that includes Named Entity recognition. Due to the well selected anchors it was found to outperform the best two sentence alignment algorithms so far published for the Hungarian-English language pair.

Keywords: sentence segmentation, sentence alignment, length-based alignment, hybrid method, Named Entity recognition, anchor, cognates, dynamic programming

1 Introduction

In the last few years parallel corpora have become evermore important in natural language processing. There are many applications which could benefit from parallel texts like (i) automatic translation programs (as machine learning algorithms) that are used as training databases, (ii) translation support tools that can be obtained from them (translation memories, bilingual dictionaries) and (iii) Cross Language Information Retrieval methods. These applications require a high-quality correspondence of text segments like sentences. Sentence alignment establishes relations between sentences of a bilingual parallel corpus. This relation may not have just a one-to-one correspondence between sentences; there could be a many-to-zero (in the case of insertion or deletion), many-to-one (if there is a contraction or an expansion) or even many-to-many alignments.

Various methods have been proposed to solve the sentence alignment task. These are all derived from two main classes: length-based and lexical methods,

*Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and University of Szeged H-6720 Szeged, Aradi vértanúk tere 1., Hungary, E-mail: {tothkr, rfarkas, kocsor}@inf.u-szeged.hu

but the most successful are combinations of them (hybrid algorithms). *Algorithms using the sentence length* are just based on statistical information given in the parallel text. The common statistical strategies all use the number of characters like Gale & Church's [8] or words like Brown et. al.'s method [1] of sentences which models the relationship between sentences to find the best correspondence. These algorithms are not so accurate if sentences are deleted, inserted or there are many-to-one or many-to-many correspondences between sentences. *Lexical-based methods* [2] [10] utilise the fact that if the words in a sentence pair correspond to each other, then the sentences are also probably translations of each other. Length-based methods align sentences quickly and the alignment is moderately accurate, while the lexical based methods are more accurate but much slower than sentence length-based alignment techniques.

Many applications combine methods which allow the generation of a fast and accurate alignment [4, 13]. These hybrid algorithms utilize various kind of anchors to enhance the quality of the alignment such as numbers, date expressions, various symbols, auxiliary information (like session numbers and the names of speakers in the Hansard corpus[1]) or cognates. Cognates are pairs of tokens of historically related languages with a similar orthography and meaning like parlament/parliament in the case of the English-French language pair. Several methods have also been published to identify cognates. Simard et. al. [20] considered words as cognates, i.e. those that had a correspondence with at least four initial letters, so pairs like government-gouvernement should be excluded. McEnery and Oakes [12] did the calculation of the similarity of two words using Dice's coefficient. These cognate-based methods work well for Indo-European languages, but languages belonging to different families (like Hungarian-English) or with different character sets the number of cognates found is low.

The newest generation of algorithms uses both the length and lexical information but they are based on the Machine Learning paradigm [3, 6]. These approaches requires a great and precise (manually labeled) training corpus which is not present for English-Hungarian at the moment.

Methods have been published for Hungarian-English language pair by Pohl [15] and Varga et. al. [23]. These are also hybrid methods that use a length-based model, but to increase the accuracy Pohl uses an anchor-finding method and the algorithm developed by Varga (called Hunalign) based on a word-translation approach.

In this paper we will introduce an efficient hybrid algorithm for sentence alignment based on sentence length and anchor matching methods that incorporate Named Entity recognition. This algorithm combines the speed of length-based models with the accuracy of the anchor-finding methods. Our algorithm here exploits the fact that Named Entities cannot be ignored from any translation process, so a sentence and its translation equivalent contain the same Named Entities. With Named Entity recognition the problem of cognate low hits for the Hungarian-English language pair can be resolved. To the best of our knowledge this work is

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC95T20

among approaches for any language pair, the first sentence alignment method that uses Named Entities as anchors.

To handle the problem of sentence alignment an efficient sentence segmentation method and an accurate parallel corpus are needed. We will introduce our expert rule based sentence segmentation and our parallel corpus as well. The recently built corpus contains over 5000 sentences per language and seeks to represent normal everyday language.

In Section 2 the sentence segmentation problem is presented, then Section 3 is devoted to sentence alignment. Section 4 introduces our reference corpus for sentence segmentation for the Hungarian-English language pair along with experiments carried out using our algorithms and several other algorithms. Our results on sentence segmentation and sentence alignment will be discussed here as well. Lastly in Section 5 we provide a short summary and some suggestions for the future.

2 The segmentation problem

The success of sentence alignment depends on the location of sentence boundaries. A common definition of a sentence is: *A sentence is a syntactically autonomous sequence of words, terminated by a sentence-end punctuation.* The term sentence-end punctuation includes full-stops ('.'), exclamation marks ('!') and question marks ('?'), but a sentence ending might be denoted by a colon (':') or semicolon (';'), provided the sentence can stand on its own syntactically (be syntactically autonomous). This definition works well if the text contains sentences in the narrowest sense. But in cases where the input contains structured elements (like tables or enumerations) this definition becomes useless because it requires that a sentence always end with a sentence-end marker. Thus we chose to redefine the meaning of a sentence from our computer linguistic perspective: *A character-stream is regarded as a sentence if it is a sentence in the narrowest sense, a title, an item of an enumeration or a cell in a table.*

Segmenting a text into sentences is a non-trivial task since all end-of-sentence punctuation marks are ambiguous. The most ambiguous sentence-end-punctuation is the full-stop. A full-stop could be a part of a date, denote an ordinal number in Hungarian, an abbreviation, be the end of a sentence, or even an abbreviation at the end of a sentence. The following sentence contains full-stops that have different roles:

```
A Szamos u. 16. alatt található XX. században épült kb. 20 méter magas épületet 2005. 06. 05. és 2006. 06. 05. között az XY. Kft. újította fel.
```

This sentence would probably be segmented into 12 sections by a sentence segmentation application that identifies a sentence boundary after each full-stop. This example and the following statistics demonstrate that the problem of sentence segmentation is worth spending some time on in order to come up with a solution. In the Brown corpus 10% of the full-stops denote abbreviations [7]. According to

[11], 47% of the full-stops in the Wall Street Journal lie inside an abbreviation and in scientific texts it is even more: from 54.7% to 92.8% [14]. Like the full-stop an exclamation mark or a question mark can be inside a sentence e.g. when they occur within quotation marks or parentheses, as in the following sentence:

```
"Látok!" - mondta a vak (aki lehet, hogy nem is vak!?)
```

To handle these problems we used the following rule based system. We collected two lists; *special characters* that are different types of quotations and parentheses, and *potential sentence-end-marks* that are full-stops, exclamation marks, question marks, colons and dots.

The algorithm has three steps:

Step 1 The first step of our segmentation process is the removal of *special characters* from the front and the end of every word.

Step 2 The word ending with a potential-sentence-end-marker (*candidate*) is analyzed: it could be an ordinal number, an abbreviation or a simple word. It has then to decide whether the candidate is an ordinal number. As for whether the word is an abbreviation or a simple word, it checks it against a look-up abbreviation list.

Step 3 The candidate's environment (the word following it) is analyzed:

(a) If there is no subsequent word, the sentence boundary has been identified.

(b) If the candidate is a simple word, and is followed by a word that is a number or begins with a capital letter, we identify a sentence boundary; otherwise there is no boundary.

(c) If the token is an abbreviation we do not segment because the abbreviation might be followed by number (like 'ca. 30'), or an abbreviation (Prof. or Dr.) or a proper name (Dr. Müller).

(d) If the candidate is an ordinal number, and it is the first token in the sentence, we do not identify a sentence boundary (but with this method we can identify rows of tables). If the ordinal number is followed by a number, or a word has a lower case first letter we do not identify a sentence boundary.

(e) A special case of the sentence-end-markers is the colon. In cases after the colon a sentence in the narrowest sense is sought: we identify a sentence boundary after a colon (as in 'Az EU alábbi intézményei a következő feladatokat látják el: Az EU Bíróság bírál.') Otherwise the colon is followed by an enumeration (like 'Halihó Malacka, vegyél nekem: mézet, kenyeret, szalonnát.') then we recognize it as one sentence.

3 The hybrid model

After the sentence boundaries are determined – using the decision process described above – for Hungarian and English we need to perform a sentence alignment in a paragraph.

Figure 1 outlines our model. As input we have two texts, a Hungarian and its translation in English. In the first step the texts will be sentence segmented, and then paragraph aligned. We look for the best possible alignment within each paragraph. For each Hungarian-English sentences we determine the cost of the sentence alignment with the help of dynamic programming. At each step we know

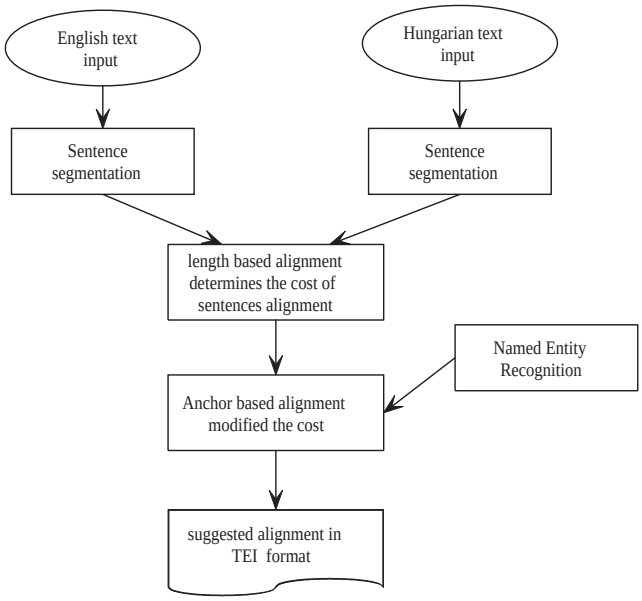

Figure 1: The overview of the alignment system

the cost of the previous alignment path, and the cost of the next step can be calculated via the length-based method and anchors (including Named Entities) – as described later in detail – for each possible alignment originating from the current point (from one-to-one up to three-to-three). The base cost of an alignment is Δ (see Section 3.1), which is increased by punishing for many-to-many alignments. Without this punishment factor the algorithm would easily choose, for example, a two-to-two alignment instead of the correct two consecutive one-to-one alignments. This base cost is then modified by the matched anchors. The normalized form of the numbers, the special characters collected from the current sentences and each matching anchor together reduce the base cost by 10%. The cost is also reduced by 10% if the sentences have the same number of Named Entities.

The problem of finding the path with minimal cost (after the cost of each possible step has been determined) is solved by dynamic programming. The search begins from the first sentences of the two languages and must terminate in the last sentences of each language text. For this we used the well known forward-backward method in dynamic programming.

3.1 Length-based alignment

This module exploits the fact that sentence lengths are correlated. The measure of a sentence length is the number of characters in a sentence, just like that in the Gale and Church [8] algorithm.

We will assume that the ratio, between the length of the source sentence and target sentence, has a normal distribution (independent and identically distributed from sentence to sentence). The mean and standard deviation of each can be calculated from our new Hungarian-English parallel corpus (introduced in Section 4.2.1): $E(l_1/l_2) \approx 1.1$ and $V(l_1/l_2) \approx 7.9$, where l_1 is the number of characters in a Hungarian sentence and l_2 is the number of characters in its translation.

Just like [8] we define δ to be $(l_2 - l_1 E(l_1/l_2)) / \sqrt{l_1 V(l_1/l_2)}$ so that it has a normal distribution with zero mean and a variance of one (at least when the two sentences in question are actually the translations equivalents of each other). The base cost of the alignment (for two sentences with length l_1 and l_2) respectively will be $\Delta = -logP\,(match|\delta(l_1, l_2))$. The log has been introduced here so that adding costs will produce desirable results.

3.2 Anchors

The published approaches for a Hungarian-English language pair judged the words containing capital letters or digits of equal amount in the text to be the most trusted anchors, but any mistakenly assigned anchors have to be filtered. Unlike other algorithms our novel method needs no filtering of anchors because the alignment works with the help of exact anchors like Named Entities. The following example illustrates the difference between using capitalized words as anchors against using Named Entities as anchors:

```
Az új európai dinamizmus és a változó geopolitikai helyzet arra
késztetett három országot, név szerint Olaszországot, Hollandiát
és Svédországot, hogy 1995.  január 1-jén csatlakozzon az Európai
Unióhoz.
```

```
The new European dynamism and the continent's changing geopolitics
led three more countries - Italy, Netherlands and Sweden - to join
the EU on 1 January 1995.
```

In the Hungarian sentence there are 5 capitalized words (Olaszországot, Hollandiát, Svédországot, Európai, Unióhoz), unlike its English equivalent which contains 7 ones (European, Italy, Netherlands and Sweden, EU, January) so using this feature as an anchor would give false results, but an accurate Named Entity recognizer could help it. This example demonstrates as well that cognates cannot be used for a Hungarian-English language pair.

Thus we suggest modifying the base cost of a sentence alignment with the help of the following anchors: special characters, the normalized form of the numbers and Named Entity recognition instead of a bilingual dictionary of anchor words or the

number of capital letters in the sentences. These result in a text-genre independent anchor method that does not require any anchor filtering at all.

3.2.1 Named Entities

Instead of using capitalized words present in the sentences we use the Named Entity Recognition module. It is used because in English more words are written with a capital letter than their Hungarian equivalents. Some examples from the Hungarian-English parallel corpus indeed demonstrate this fact:

- I (én) personal pronoun
- Nationality names: ír söröző = Irish pub
- Location terms: Kossuth Street/Road/Park
- When repeating an expression, the expressions become shorter: pl: European Union = Unió
- Names of countries: Soviet Union = Szovjetunió
- The names of months and days are written with capital letters.

The identification and classification of proper nouns in a plain text is of key importance in numerous natural language processing applications. It is useful in sentence alignment because Named Entities cannot be ignored in any translation process, so a sentence and its translation equivalent contains the same number (and types) of Named Entities. As far as we know our work is the first sentence alignment method for a language pair that uses Named Entities as anchors.

A slightly modified version of the multilingual Named Entity recognition system described in [22] was used here in this work. This system (which appears to be currently the only statistical Named Entity recognition for Hungarian) achieved an accuracy[2] of over 98.7% on unknown documents in Hungarian (Szeged NE corpus [21]) and 97% for documents in English (CoNNL 2003 shared task [18]). The main aim of [22] was to recognize Named Entities and place them into one of four classes (person, organization, location and miscellaneous). The accessible tagged datasets concentrated on the business domain, unlike our parallel texts which dealt with a wide range of domains. Because of the lack of a suitable training corpora we chose an easier problem, namely recognizing Named Entity phrases (a multiword chain) without classification.
Our statistical approach worked as follows:

1. It extracts features from a tagged train corpora. We collected various types of numerically uncodable information describing each term and its surroundings. A subset of the features used tried to capture the orthographical regularities of proper nouns like capitalization, inner-word punctuation and so on. Another

[2]considering the two class (named entity/non named entity) phrase level evaluation metric

set of attributes described the role of the word and its neighboring words in the sentence. The remaining parameters were various lists of trigger words and ratios of capitalized and lowercase words in large corpora.

2. In this way the problem could be treated as a supervised (more precisely, a two-class classification) task. The C4.5 decision tree [16] with pruning and AdaBoost [19] after 30 iterations was trained on Hungarian and English texts. Different models were learned for the two languages but they were based on the same feature set.

3. The learned models tagged the Named Entity phrases in the input parallel texts. Because the correct tagging was not known we could not measure the accuracy of this tagging, but the experiments described in Section 4.2 revealed that it was definitely helpful.

3.2.2 Special characters

We used the special characters in the sentences like %, §, $, @, & as anchors, because they may be present in the source language and a target language sentence in the same form. Other special characters are used as anchors in the literature as well (like quotation, exclamation mark, question mark [9]), but they were not included because they can confuse the program in the Hungarian-English alignment task.

Take the following examples:

```
angol = I wish I had a bike.
magyar = Bárcsak lenne egy biciklim!
```

The Hungarian exclamation mark is usually (90% in our Hungarian-English parallel corpora) replaced by a full-stop in its English counterpart. As the next example shows, there are differences in the usage of the apostrophes and quotations in the English and Hungarian sentences. In most cases Hungarian quotations have an apostrophe equivalent in an English sentence, and vice versa.

```
"Tibi!" - mondtam az uramnak.

'Tibi!'  - I said to my husband.
```

3.2.3 Normalized form of numbers

Our efficient sentence alignment method treats the normalized form of Arabic or Roman digits. During the normalisation of the digits all characters that are not digits are deleted then we get a digit in a normalized form. With this method we got a language independent form (1.2 = 1,2) that can be compared during the alignment process.

4 Experiments

In this section our results on sentence segmentation and on sentence alignment are presented, and the reference algorithms and corpora are given.

4.1 Sentence segmentation

Here the reference corpora for Hungarian and English and the baseline algorithms are introduced for evaluating our expert rule-based sentence segmentation approach.

4.1.1 Reference corpora

The test corpus for Hungarian sentence segmentation was compiled from sentences that came from three subcorpora of the Szeged Treebank[3] [5] (Népszabadság, Népszava and Heti Világgazdaság). The first two subcorpora were chosen here because a sentence segmentation algorithm will be used on general texts and these truly mirror everyday language. The third is written in business language, and can be used for testing our algorithm on a harder text genre.

The English sentence track evaluation was carried out on the Wall Street Journal Corpus[4], which contains articles from the Wall Street Journal, and consists of 5000 randomly selected sentences. We choose this corpus because we wanted to test our algorithm on articles similar to Hungarian ones, and it was written in normal everyday language.

4.1.2 Baseline algorithms

We compared our algorithm against two algorithms. The first was the baseline algorithm that labels each punctuation mark as a sentence boundary. The description of the second one – called Huntoken – has not yet been published, but some of its results has been used in our three subcorpora[17].

4.1.3 Results

To assess the quality of sentence segmentation precision, recall and F-measure scores of correct segmentations were used.

Tables 1 below list the results of the segmentation on the three subcorpora compared with the results of the first baseline algorithm. We could not compare our results in such detail with Huntoken because only the precision scores have been published so far.

Our expert rule-based algorithm performs significantly better on all subcorpora than the baseline algorithm. These experiments highlight the effect of meaning

[3]The Szeged Treebank is a manually annotated natural language corpus. This is the largest manually processed Hungarian database that can be used as a reference material for research in natural language processing

[4]http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2000T43

Algorithm	Correct	Incorr.	All f.	Etalon	prec.	rec.	$F_{\beta=1}$
Baseline	6450	697	7147	7797	0.9009	0.8216	0.8593
Expert Rule b.	7781	17	7798	7797	0.9980	0.9981	0.9980

Table 1: The three subcorpora together

differences of potential punctuation marks as well. The baseline algorithm achieved poor precision and recall scores on the Heti Világgazdaság corpus (which contains economic texts) but our algorithm gave much the same results as those for the two other corpora. This is probably due to the higher frequency of abbreviations, ambiguous sentence boundaries and special punctuation marks.

Table 2 lists the precision scores of the three algorithms. Our results turned out to be similar to those published in [17]. The difference can be said to be significant only on the economy texts (a 66% error reduction)

Algorithm	Népszabadság	Népszava	Heti Világgazdaság	all
Baseline	0.9133	0.9113	0.8780	0.9009
Huntoken	0.9976	0.9977	0.9937	0.9963
Expert Rule based	0.9976	0.9985	0.9979	0.9980

Table 2: Precision of the three sentence segmentation algorithm

To evaluate the English text, the first baseline algorithm were used. With the English test corpora the baseline method performed very badly, but our algorithm kept the error rate below 1% (see Table 3). The reason for this is that in this corpus there were a lot of parentheses and quotation marks in the words, and there were also quite a few abbreviation and ordinal numbers.

Algorithm	Correct	Incorr.	All f.	Etalon	prec.	rec.	$F_{\beta=1}$
Baseline	3152	3257	6409	5021	0.4918	0.6278	0.5515
Expert Rule b.	4972	38	5010	5021	0.9924	0.9902	0.9913

Table 3: English results

These results demonstrate that our effective sentence segmentation algorithm generates errors of 1% or less on both Hungarian and English texts. This achievment means that our approach is competitive with the best published results for Hungarian and English to date.

4.2 Sentence alignment

Soon we will discuss the results of experiments on our alignment algorithms. But first we need to elaborate on the built corpora and two baseline algorithms from Hungarian literature.

4.2.1 The corpora for sentence alignment

Parallel corpus Currently, there are two sentence-level-aligned Hungarian-English parallel corpora at our disposal. One of them is the so-called Orwell corpus[5] that is based on Georg Orwell's novel 1984 and the other one is a Hunglish corpus [6]. These corpora often contain special words, phrases and jargon, that is why we decided to build our own corpus.

With high quality translation and representability in mind, in the course of Hungarian-English parallel corpus building the following texts were collected:

- **Language book sentences**: This subcorpus includes detached parallel sentences from Dévainé Angeli Mariann's *Angol nyelvtani gyakorlatok* and Dohár Péter's *Kis angol nyelvtan.* These books were compiled for students preparing for a language exam and therefore their wording is not very realistic. There are sentences which truly represent present-day English but, at the same time, there are some overly artificial, 'fabricated' sentences too. These books were written to represent the characteristics of English and not present-day parlance. This subcorpus currently contains over 5000 sentences.

- **Texts on the EU**: These texts were gathered from an official EU website http://europa.eu.int. Under the title *Europe in 12 lessons* there are 13 general descriptions about the EU. This subcorpus is a general Hungarian-English text collection.

- **Bilingual magazines**: This subcorpus is comprised of articles taken from the magazines of Malév Horizon and Máv Intercity.

- **Speech corpus of the Multext-East**: The Multext-East corpus consists of 40 items of 5-sentence long units. The 5 sentences of a unit are correlated and they are available in written form in both Hungarian and English. Text units include topics written in everyday parlance, tell one how to order a taxi, find a restaurant, or call a customer service end so on.

Named Entity training corpora. To train our model on Hungarian texts, we used a a sub-corpus of the Szeged Treebank [21] where the correct classification of Named Entities had also been added[7]. It contains business news articles taken from 38 NewsML topics (9600 sentences) ranging from acquisitions to stock market changes or the opening of new industrial plants.

The Named Entity system for English was trained on a sub-corpus of the Reuters Corpus, consisting of newswire articles from 1996 provided by Reuters Inc. (–the shared task of the CoNLL 2003 Named Entity challenge). It contains texts from domains ranging from sports news to politics and the economy.

[5] http://nl.ijs.si/ME/CD/docs/1984.html
[6] http://mokk.bme.hu/resources/hunglishcorpus
[7] Both Hungarian and English datasets can be downloaded free of charge for research purposes.

4.2.2 Reference alignment methods

Hunglish, translation- and length-based alignment In the first step the algorithm loads the English-Hungarian dictionary that was based on a unified version of the Vonyó and Hókötő dictionaries[8]. The first step of the aligning algorithm provides a rough translation of the Hungarian sentence by substituting each word with its most frequently occurring dictionary translation or, when absent, with the word itself. Then this rough translation is compared, sentence by sentence, with the actual target text.

The similarity rate between sentences is found by looking at the number of mutual occurrences (the very frequent words having been removed from both the raw translation and the original English text) and the sentence length which is measured in characters, but the algorithm also specifically recognizes numbers written in numerical form. The task is then solved with the help of dynamic programming methods [23].

Length- and anchor matching-based alignment of Pohl The other sentence-synchronizing algorithm was worked out by Pohl [15]. The implemented algorithm was built on Gale & Church's sentence-length alignment, and it also included dynamic programming techniques to determine the sentences to be aligned. The only real difference from the original algorithm was that it had to take into consideration the cost of anchor-synchronisation when calculating the overall costs. When running it uses a heuristic method to calculate the gain, which helps it to recognize sentence insertions and deletions in the text. The gain is defined here as follows. It is the number of common anchors in text units divided by the total number of anchors in the text units, then this fraction is divided by the number of text units involved. Pohl regards on the other hand the number of words containing numbers or capital letters as the most reliable anchors. He employed the method published by Ribeiro et. al. to filter out the mistaken anchors. It defines two statistical filters, both of which apply a linear regression margin calculated on the basis of the anchor-candidate's position in the text. In the first step the points outside a certain range – determined using an adaptive histogram-based filter applied around the linear regression margin – were disregarded, then the points outside the confidence bracket of the regression margin were found.

4.2.3 Results

Our hybrid algorithm was compared with Pohl's length- and anchor matching-based one and with the Hunglish's dictionary- and sentence length-based hybrid ones. Pohl's algorithm also had to be reimplemented. The comparison was not complete, but it used just one-to-one, two-to-one and one-to-two alignment types.

The first row shows what kind of alignments are possible in the reference alignment, like one-to-one, one-to-two or many-to-many. There is no one-to-zero alignment in our parallel corpus even through there could be. The second row shows

[8] http://almos.vein.hu/ vonyoa/SZOTAR.HTM

	1:1	1:2&2:1	2:2	N:M
suggested alignment	4875	415	0	0
correct of sugg. align.	4556	165	0	0

Table 4: Pohl's Results

how many of these alignment types were found by Pohl's algorithm, and the last row shows how many of the suggested alignments were correct. Table 5 gives the corresponding results for our algorithm, which, as the reader will notice, are not so different.

	1:1	1:2&2:1	2:2	N:M
suggested alignment	4957	339	3	1
correct of sugg. align.	4698	252	1	0

Table 5: Our Results

The algorithm of Pohl's chose one-to-two and two-to-one alignments with a poor precision (just 39%). Our hybrid algorithm on the other hand was more accurate in these cases and it even handles two-to-two and n:m alignments as well. In the one-to-one alignment task they achieved similar results. Our algorithm was better here as well, but this is probably only due to Pohl choosing too many one-to-two alignments instead of more one-to-one alignments.

Table 6 summaries the results of the three hybrid methods. Precision and recall are the commonly accepted metrics for evaluating the quality of a suggested alignment with respect to a test corpus. We employ the F-measure here as well, which combines these metrics into a single efficiency measure:

$$precision = \frac{number\ of\ correct\ alignments}{number\ of\ proposed\ alignments}$$

$$recall = \frac{number\ of\ correct\ alignments}{number\ of\ reference\ alignments}$$

$$F_{\beta=1} = 2\ \frac{recall * precision}{recall + precision}$$

Algorithm	Precision	Recall	$F_{\beta=1}$
Pohl hybrid	0.9016	0.9016	0.9016
Hunalign	0.8993	0.9786	0.9370
NE-based	0.9341	0.9456	0.9398

Table 6: Results of Hungarian hybrid methods

The high recall of the hybrid dictionary-based method is largely due to the dictionary (it offers a huge number of one-to-one alignments), but it did not attain

a 90% precision score. Contrary to our algorithm, it has a recall and precision of over 90 % thanks to the good choice of anchors.

After a manual analysis, we found that the bigger part of the errors come from paragraphs where there are not any anchor (neither Named Entities, numbers nor punctuation) in the sentences. On the other hand the recognition of Named Entities is far from perfect, its error is propagated to the alignment. If a larger and more general Named Entity training corpus will be available a more accurate recogniser model could be trained and different types of entities could be used which could further improve our results.

Viewed overall, our new hybrid algorithm is approximately 4% better than the approach which inspired our study (Pohl's anchor matching based algorithm) and it achieved slightly better results than those for Hunalign. The real advantage over Hunalign is its speed of alignment. We used a very fast (in alignment time) Named Entity recognizer that did not need to search through a huge database dictionary.

5 Conclusions and future work

In this paper we introduced a language independent, expert rule-based sentence segmentation method (which we found has a typical error rate of $< 1\%$), a Hungarian-English parallel corpus containing everyday language – which was designed for machine translation – and a novel Named Entity-based hybrid sentence alignment method (the first step of machine translation) that combines accuracy (a roughly 6% error rate) with speed.

The results of the previous section demonstrate that our system is competitive with other sentence alignment methods published for the Hungarian-English language pair. The reason for our good results is that, with the help of Named Entity recognition, more anchors can be matched so the problem of low hits of the cognate pairs for a Hungarian-English language pair is effectively solved. The use of multilingual Named Entity recognition systems also provides a way of finding appropriate anchors for language pairs even when they belong to distinct language families.

In the future it would be useful to build and to learn on a Named Entity corpus that incorporates everyday language. Then the advantage of using a Named Entity classifier would probably become apparent and it should improve the precision of Named Entity anchors. In addition, we would like to test our system on diverse text sources to see how well it performs.

References

[1] Brown, P. F., Lai, J. C., and Mercer, R. L. Aligning sentences in parallel corpora. In *Meeting of the Association for Computational Linguistics*, pages 169–176, 1991.

[2] Chen, S. F. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 9 – 16. Columbus, Ohio, 1993.

[3] Chuang, Thomas C. and Chang, Jason S. Adaptive bilingual sentence alignment. In *AMTA '02: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, pages 21–30, London, UK, 2002. Springer-Verlag.

[4] Collier, N., Ono, K., and Hirakawa, H. An experiment in hybrid dictionary and statistical sentence alignment. In *COLING-ACL*, pages 268–274, 1998.

[5] Csendes, D., Csirik, J., and Gyimóthy, T. The szeged corpus: A pos tagged and syntactically annotated hungarian natural language corpus. In *Proceedings of the 7th International Conference on Text, Speech and Dialogue (TSD 2004)*, pages 41–47, 2004.

[6] Fattah, Mohamed Abdel, Ren, Fuji, and Kuroiwa, Shingo. Probabilistic neural network based english-arabic sentence alignment. In *CICLing*, pages 97–100, 2006.

[7] Francis, W. and Kucera, H. *Frequency analysis of English usage: Lexicon and grammar*. Houghton Mifflin, Boston, 1982.

[8] Gale, W. A. and Church, K. W. A program for aligning sentences in bilingual corpora. In *Meeting of the Association for Computational Linguistics*, pages 177–184, 1991.

[9] Hofland, K. and Johansson, S. The translation corpus aligner: A program for automatic alignment of parallel texts. In Johansson, S. and Oksefjell, S., editors, *Corpora and Cross-linguistic Research: Theory, Method, and Case Studies*, pages 87–100. Amsterdam: Rodopi, 1998.

[10] Kay, M. and Röscheisen, M. Text-translation alignment. volume 19, pages 121–142, 1993.

[11] Liberman, M. Y. and Church, K. W. Text analysis and word pronunciation in text-to-speech synthesis. In *Sadaoki Furui and Man Mohan Sondhi, editors, Advances in Speech Signal Processing*, pages 791–831. Marcel Dekker, Inc., 1992.

[12] McEnery, A. M. and Oakes, M. P. Cognate extraction in the crater project. In *S. Armstrong-Warwick and E. Tzoukerman (eds.), Proceedings of the EACL-SIGDAT Workshop (Dublin)*, pages 77 – 86, 1995.

[13] Moore, R. C. Fast and accurate sentence alignment of bilingual corpora. In *AMTA '02: Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, pages 135–144, London, UK, 2002. Springer-Verlag.

[14] Muller, H., Amerl, V., and Natalis, G. *Worterkennungsverfahren als Grundlage einer Universalmethode zur automatischen Segmentierung von Texten in Sätze. Ein Verfahren zur maschinellen Satzgrenzenbestimmung im Englischen. Sprache und Datenverarbeitung, 1.* 1980.

[15] Pohl, G. Szövegszinkronizációs módszerek, hibrid bekezdés– és mondatszinkronizációs megoldás. In *Proceedings of Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2003)*, pages 254–259, 2003.

[16] Quinlan, J. R. *C4.5: Programs for machine learning.* Morgan Kaufmann, 1993.

[17] és Kommunikáció Tanszék Média Oktató és Kutató Központ, BME Szociológia. Hunglish cd-rom, http://szotar.mokk.bme.hu/hunglish/search/corpus. 2006.

[18] Sang, E. F. Tjong Kim and Meulder, F. De. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Daelemans, Walter and Osborne, Miles, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.

[19] Schapire, Robert E. *The Strength of Weak Learnability*, volume 5. 1990.

[20] Simard, M., Foster, G., and Isabelle, P. Using cognates to align sentences in bilingual corpora. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine translation (TMI92), (Montreal)*, pages 67–81, 1992.

[21] Szarvas, Gy., Farkas, R., Felfoldi, L., Kocsor, A., and Csirik, J. A highly accurate named entity corpus for hungarian. In *Proceedings of LREC2006*, 2006.

[22] Szarvas, Gy., Farkas, R., and Kocsor, A. *A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms.* Springer-Verlag, 2006.

[23] Varga, D., Halacsy, P., Kornai, A., Nagy, V., Nemeth, L., and Tron, V. Parallel corpora for medium density languages. pages 590 – 596. Borovets, Bulgaria, 2005.