English-Hungarian NP Alignment in MetaMorpho TM

Gábor Pohl

Faculty of Information Technology
Péter Pázmány Catholic University
H-1083 Budapest, Práter utca 50/A, Hungary
pohl@itk.ppke.hu

Abstract. In this paper, a fast automatic NP alignment technique developed for MetaMorpho TM is presented. MetaMorpho TM is an EBMT-based translation memory that stores not only full sentence pairs but also NP pairs in its database of translations. In order to fulfill speed requirements of a translation memory (segments have to be stored quickly), in the proposed NP alignment algorithm time consuming statistical data collection is substituted for stemmed lexical matching using a bilingual dictionary, cognate matching and POS matching. A simple heuristic means of extracting Hungarian NP candidates without a deep parser is also presented in this paper. Parsed NPs of an English sentence are mapped to the words of the Hungarian translation and the shortest span containing all matched words is expanded to a full Hungarian NP using simple rules. The first experiment shows that high precision can be reached using the two algorithms discussed.

1 Introduction

Noun phrase (NP) alignment is the process of identifying corresponding NP pairs in human translations. In this paper, automatic NP alignment techniques developed for our EBMT-based translation memory (TM), MetaMorpho TM (Grőbler et al., 2004; Hodász & Pohl, 2005), will be presented and analyzed.

MetaMorpho TM differs from the mostly language independent commercially available TM products in the following features. (1) In order to find the stored segment most similar to the one searched, a morpho-syntactic similarity measure is applied instead of simple character-based similarity measures. (2) MetaMorpho TM is a simple EBMT[1] system (Nagao, 1984; Somers, 2003). Unlike traditional TM products not only whole sentences are searched in the memory. NPs, and the sentence skeleton (derived from the sentence by substituting NPs with symbolic NP slots) are also searched in the database, and their most probable translations are morphologically altered and combined to form a possible translation sentence. Hence the recall of the translation memory is increased by suggesting translations built up from NP and sentence skeleton translations looked up separately. If the translation of words or phrases in the NPs or the sentence skeleton depend on segments looked up separately, this approach can produce incorrect translations, but even in such cases, the suggested translation might contain well-translated parts that can help the human translator. We think the achieved higher recall justifies the slightly lower precision.

In the MetaMorpho TM system, not only full sentence translations but also NP and sentence skeleton translations have to be stored in the database of translation equivalents. NPs of the stored sentence pair have to be aligned either by the translator or by an automatic means. Leaving the tedious task of NP alignment to the translator would decrease productivity, thus an automatic means of NP alignment was developed for our TM. NP alignment is considered to be a major error source in our translation memory, so we had to find a precise NP alignment technique.

[1] Example Based Machine Translation

Because available means were not suited to the English-Hungarian language pair and the speed requirements of a translation memory, we developed a simple dictionary-based algorithm to measure NP-NP similarity, and we carried out a successful experiment with our new means of determining the Hungarian correlates of English NPs without using a Hungarian parser.

2 Automatic NP-alignment

In the automatic noun phrase alignment process, noun phrases that are considered each other's translations (in a bilingual sentence pair) are mapped to each other by an algorithmic means. The mappings may be one-to-one or one-to-zero. Noun phrases may remain unmatched because (1) the two languages use different syntactic structures (see example E1 and E2 below); (2) there are concepts that are expressed with different phrase types in the two languages (see example E3 below); (3) the translator may rephrase the translation, changing words or even phrases (see example E4 and E5 below).

NPs must remain unmatched if they have no exact translation. By exact translation we mean semantic equivalence, minor syntactic differences like different usage of pronouns and determiners are allowed. Mapping of only partially matching NPs to each other should be considered an error in the automatic alignment (e.g. mapping *floppy disk drive* to *lemezmeghajtó,* the Hungarian translation of *disk drive* is not acceptable).

[I] have read [his newbook].
Elolvastam [az új könyvét]. ("I_have_read [the new book_his]") (E1)

Example 1: Personal pronouns of English may only be mapped to personal suffixes of verbs in Hungarian. The English pronoun *I* has no NP translation in the Hungarian sentence.

there is [no data loss]
nincs [adatvesztés] ("there_is_not [data_loss]") (E2)

Example 2: In the Hungarian phrase the verb is negated and not the NP.

[Rob] had [a huge breakfast].
[Rob] jól bereggelizett. ("[Rob] had_breakfast well") (E3)

Example 3: The English verb phrase *have breakfast* is usually translated to a single Hungarian verb *reggelizik,* hence the English phrase *have a huge breakfast* has no NP translation in the Hungarian sentence.

[Tom] ate [ice-cream].
[Tom] [fagyit] evett. ("[Tom] [ice-cream+ACC] ate."
[Tom] fagyizott. ("[Tom] ice-cream_ate") (E4)

Example 4: *Ice-cream eating* can be expressed in Hungarian with a phrase similar to the English one or with a simple Hungarian verb (*fagyizik*), although there is a slight semantic difference between the two forms (the first Hungarian translation may express the fact that Tom ate ice-cream and not some other type of food.) If the translator chooses the single verb translation, no NP translation of *ice-cream* can be found in the Hungarian sentence.

If you don't do this…
Ha ez nem történik meg… ("If it does not happen…") (E5)

Example 5: The translator—following a style guide—may use different phrases in the translation.

2.1 Previous work

Previous and related works include corpus-based statistical phrase alignment methods, and parse tree alignment techniques for EBMT systems. Recent advances in tree alignment (e.g. Groves, 2004) are promising but our English and Hungarian parses are too different for such methods depending on the internal structure of NP trees. Corpus-based statistical means, like the NP chunk aligner developed by Kupiec (1993) and recent phrase alignment techniques (de Gispert & Marino, 2006) are more robust, but they require reprocessing of the whole translation memory after a new sentence pair is stored. In a TM product, a new sentence pair should be stored in less than one second, because the translator expects a segment of translation to be brought up even if it were stored with the previously translated sentence. Ordinary translation memories also store translations quickly and they are immediately able to look up such segments. Hence reprocessing of the whole memory was not viable in MetaMorpho TM. Statistical methods suffer given the number of different word forms, i.e. the rich morphology of Hungarian. Applying a stemmer would significantly reduce the number of surface forms, but selecting the appropriate stem of each word would not be easy.

Therefore, in our NP aligner we substituted the time consuming statistical data collection for dictionary usage.

2.2 Dictionary-based NP alignment

Developing a new NP aligner algorithm precision and speed were our main goals. Precision was more important than recall, because poorly aligned NP pairs stored in the memory would incorrectly appear as suggested translations until they are purged from the memory, but purging such translations is still a time consuming task in our TM. (Later on we plan to develop methods to filter rarely accepted translations.) Hence, for the time being, we preferred higher precision, but we wanted to develop a means with adjustable precision/recall rates.

To achieve high speed, we decided to avoid the computationally intensive corpus processing, and developed a solution based on (1) fast, stemmed lexical matching using a bilingual dictionary, (2) cognate matching (Simard et al, 1992), and (3) POS-matching.

Our heuristic NP aligner algorithm calculates a heuristic matching score for all possible English-Hungarian NP pairs, and marks NPs as translations of each other if and only if their matching score exceeds a threshold value and both NPs of the pair match their mate with a higher score than any other NP in the translation. The latter stipulation helps us find the best matching pairs of possible ones, if there is a best match. In reality after applying the threshold to filter candidates, conflicting pairs only remain if the sentences contain recurring NPs or NPs that are slight modifications of others in the same sentence.

2.3 NP–NP matching score

For an NP pair, we aim to calculate a heuristic scalar value that describes the similarity of the two NPs of the pair. We call this value matching score. During the calculation process, words (tokens) of the two NPs are matched using different word-matching methods. Then the similarity score is calculated by a simple formula (see F1 below) depending on the number of words matched by the different methods.

First dictionary matching is done then cognates are searched among the unmatched words. Finally POS matching is calculated among the previously unmatched words. If any function word remains unmatched, it is discarded with a small penalty in the matching score.

During the dictionary-based matching, all possible stems of words in an English NP are looked up in the dictionary using a stem index. The dictionary index also contains references to phrases (or multi-word lexemes). Longer matches are preferred, so locally longest matching phrases are selected for each word position. This way if *hard disk drive* is in an English NP and also exists in the dictionary, the shorter matching possible dictionary entries, *hard disk*, *disk drive*, *hard*, *disk*, and *drive* are not matched to words in *hard disk drive;* but an independent occurrence of the shorter entries can still be selected if they occur at some other position in the NP.

A dictionary entry found in an English NP is considered to be found in a Hungarian NP if at least one possible stem of all tokens of the entry can be matched to an unmatched token of the Hungarian NP.

Cognates are searched among the words of the English and Hungarian NPs in order to find named entities that are not in the dictionary. In our implementation two words are considered cognates if they are both longer than one character; both contain at least one capital letter, number, or other special character; and they are exactly the same word or at least their first 4 characters are the same. Later on this test may be substituted for a Levenshtein-distance (1965) based cognateness test.

By matching the part of speech tags of words not found in the dictionary, the recall of the alignment can be increased. In our experiment, we used only basic POS categories like noun, verb, adjective, preposition, determiner, pronoun, etc.

Elimination of unmatched function words is important because there are differences in function words of English and Hungarian. In Hungarian different cases are used instead of prepositions of English, and English possessive pronouns usually correspond to a determiner and some case marking on the possessed Hungarian noun. Handling unmatched words without any distinction would result in lower recall, e.g. the following NP pair could not be aligned if we had given the same penalty for all unmatched words:

- *my book* [PRON N]
- *a könyvem* ("the book-my") [DET N].

After the previously described matching techniques are carried out, the m matching sore is calculated by the following formula (F1):

$$m = \frac{a \cdot D + b \cdot C + c \cdot P - d \cdot F}{W - F}, \qquad \text{(F1)}$$

where D is the number of dictionary matched words, C is the number of matched cognates (not matched before), P is the number of words (not matched before) with matching POS tags, F is the number of remaining function words, and W is the number of all words. All values are calculated taking into account both the English and the Hungarian NPs. The constant coefficients (a=1.0, b=0.9, c=0.3, d=0.1) are guessed values, later on they are going to be trained on an NP-aligned parallel corpus. The precision/recall ratio of the alignment algorithm can be tuned by changing the coefficients or the threshold value (a value of 0.75 was applied in our first experiments).

3 NP extraction

In our first experiment, NPs of the stored sentence pair were extracted by the MetaMorpho English and a Hungarian parser (Prószéky 2006). At the time of our first experiments the Hungarian grammar was only partially implemented and had low recall and precision. Therefore, we developed a simple heuristic means of extracting Hungarian NP candidates by mapping the words of the English NPs to the words of the Hungarian sentence. Each content word of an English NP is mapped to all possible word positions in the Hungarian sentence using dictionary matching of word stems and cognate matching (described previously in section 2.3). Function words of English NPs are not mapped to Hungarian words because they are likely to occur independently as translations of other words or syntactic structures.

Matched words can occur more than once in the Hungarian sentence. The shortest span in which all the matching terms are found is selected as an NP skeleton in the Hungarian sentence. Selecting the shortest span is a computationally intensive task, but the searching space can be reduced by limiting the matching length. (In our experiments we limited the matching length, i.e. the length of possible NPs, to 10 words.)

After selecting the matched words (NP skeleton) in the Hungarian sentence, unmatched words of the English NP are matched to the unmatched words between matched Hungarian words if their POS categories are "compatible". Later on the Hungarian NP may be expanded to the left (preferably) or to the right depending on the POS of unmatched words and a basic Hungarian NP grammar. (The grammar contains simple rules, as determiners on the left side of the NP are considered part of the Hungarian NP even if they had no counterpart in the English NP.) After "guessing" the Hungarian NPs, we

calculate English–Hungarian NP matching scores.

The described method sometimes makes mistakes introduced by the ambiguity of dictionary mappings and the simple heuristics applied. Fortunately the incorrectly identified Hungarian NPs can be usually filtered by the matching score based filtering (described in section 2.2)

In our current implementation, we search Hungarian NP candidates independently. Therefore overlapping Hungarian NPs might be identified in the Hungarian sentence. However, the probability of both overlapping NPs reaching the necessary similarity score is low. Anyway, if two overlapping NPs survive the similarity threshold filtering, we discard both of them.

In the future we will see if a more sophisticated solution is not possible. One possible solution would be if words of NPs reaching the necessary matching score were marked "occupied" (from left to right), so they could not be covered in the process of expanding another NP skeleton. Using this technique showed higher recall, and only slightly less precision, but we did not have enough testing data to be sure of it.

4 Results

The first experiments with the new NP aligner and Hungarian NP guesser algorithms were carried out on a small part of the SZAK corpus (Kis & Kis, 2003) containing books on software products. We selected 40 sentence pairs with no quoted screen text and good MetaMorpho English NP parses (e.g. *Disk Management* and *Remove Mirror* are quoted screen texts in the following sentence: '*In Disk Management, right-click one of the volumes in the mirrored set and then choose Remove Mirror.'*).

The 40 English sentences contained 179 NPs and had the average length of 23 words. We hand-aligned the English NPs with their Hungarian correlates. 83 English NPs had no alignable Hungarian translation, i.e. only 56% were alignable.

For testing the alignment algorithm we used a small bilingual dictionary containing 116,000 word and phrase mappings.

Alignment precision was 84% (69 good and 13 bad alignments). Precision could be increased to 91% if the translator marked the sentences where more than half of the NPs were not alignable by hand. (In a TM product the translator could mark such sentences by checking a checkbox before storing the sentence pair.)

Recall was 65% among the hand-alignable NPs (69 good alignments of the 116 possible)

The speed of the implemented algorithms is decent, the Hungarian NP guessing and NP alignment of the longest test sentence pairs always takes less than 15 milliseconds on an average PC; which is negligible compared to the time English parsing takes.

5 Further work

Currently we are building a larger hand-aligned test corpus, that will be used to carry out a more robust evaluation of our algorithms, and when the corpus is large enough, separate parts of it will be used to fine-tune the algorithms, especially the coefficients of the matching score formula (F1), and the way guessed Hungarian NPs are expanded.

The MetaMorpho Hungarian parser improved a lot recently, so we are going to compare the results of using the parser to the ones using our NP guesser algorithm.

In the near future, as the evaluation of the MetaMorpho TM (Hodász, 2006), as a whole, continues, the effects of alignment errors, i.e. incorrectly aligned NPs and sentence skeletons stored in the memory, will be measured. Having seen the results we will decide how to deal with incorrectly stored segments.

To improve the recall of the NP alignment algorithm, we plan to extend the dictionary with pairs automatically extracted from the segment pairs already stored in the translation memory.

References

de Gispert, A., & Marino, J. B. (2006). Linguistic Knowledge in Statistical Phrase-Based Word Alignment. *Natural Language Engineering 12(1)*, 91–107.

Grőbler, T., Hodász, G., & Kis, B. (2004). MetaMorpho TM: A Rule-Based Translation Corpus. In *International Conference on Language Resources and Evaluation*, Lisbon.

Groves, D., Hearne, M., & Way, A. (2004). Robust Sub-Sentential Alignment of Phrase-Structure Trees. In *Proceedings of The 20th International Conference on Computational Linguistics, (COLING'04)*, August 2004, Geneva, Switzerland.

Hodász, G. (2006). Towards a Comprehensive Evaluation Method of Memory-Based Translation Systems. *in this volume*

Hodász, G., & Pohl, G. (2005). MetaMorpho TM: a linguistically enriched translation memory. In *International Workshop, Modern Approaches in Translation Technologies* (eds: Walter Hahn, John Hutchins, and Cristina Vertan,), (pp. 26–30) Borovets, Bulgaria.

Kis, Á., & Kis, B. (2003). A Prescriptive Corpus-based Technical Dictionary. In *Papers in Computational Lexicography: Proceedings of COMPLEX 2003*. Research Institute for Linguistics, Hungarian Academy of Sciences, Budapest.

Kupiec, J. (1993). An Algorithm for Finding Noun Phrase Correspondences in Bilingual Corpora. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics* (pp. 17–22).

Levenshtein, V. I. (1965). 'Binary codes capable of correcting deletions, insertions and reversals', Doklady Akademii Nauk, SSSR 163(4) pp. 845-848 (Russian), also Soviet Physics Doklady 10(8) pp. 707–710 (English translation).

Nagao, M. (1984). A framework of a mechanical translation between Japanese and English by analogy principle. In A. Elithorn & R. Banerji (Eds.): *Artificial and human intelligence* (pp. 173–180). Amsterdam, North-Holland.

Prószéky, G. (2006). Translating While parsing. In M. Suominen et al. (Eds.): *A Man of Measure* (pp. 449-459). The Linguistic Association of Finland, Turku

Simard, M., Foster, G. & Isabelle, P. (1992). Using Cognates to Align Sentences in Bilingual Corpora. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine translation, (TMI92)*. (pp. 67–81), Montreal.

Somers, H. (2003). An Overview of EBMT. In M. Carl. and A. Way. (eds.) *Recent Advances in Example-based Machine Translation* (pp. 3–57), Kluwer Academic Publishers, Dordrecht, The Netherlands.