[*Translating and the Computer 23*: papers from the Aslib conference held on 29 & 30 November 2001]

Improving Automatic Alignment for Translation Memory Creation

Kirsty Macdonald
UMIST, UK
kirsty.macdonald@sap.com[1]

Abstract: Currently available TM systems usually include an alignment tool to create memories from existing parallel texts. However, the alignments proposed are rarely reliable enough to allow the newly created TM to be exploited without being checked by a human user. This paper will describe a series of experiments using the popular TRADOS WinAlign program with a collection of German-English parallel texts totalling roughly 80 000 words and will look at:

- the accuracy of the proposed alignments
- which are the misaligned segments and why
- designing a misalignment checking tool

Introduction

Translation Memory (TM) programs facilitate the exploitation of previous translations, which in repetitive domains such as technical documentation are viewed as a valuable resource. Recycling 'old' translations not only saves companies both time and money but also relieves translators of repetitive work freeing up their time for other important tasks. Translation Memory programs crucially use alignment[2] programs to enable parallel corpora (previous source language and translated texts) to be loaded into the *"memory".*

The most time consuming and painstaking step of the alignment process for TM creation is checking the proposed alignments for mismatches and correcting them. In this paper the problems which arise in the automatic alignment of parallel texts for the creation of translation memories are addressed. It is the aim of this paper to give suggestions as to how the work of the translator or technical support staff correcting automatic alignments could be lessened.

This paper looks more closely at the causes of misalignments and goes some way to proposing methods for reducing these misalignments. Such methods are an analysis of factors which contribute to the poor alignments as well as a proposal for a possible tool which would find and highlight misalignments for the checker, thus substantially decreasing the time and concentration that the checking process entails.

A Definition of Alignment

Alignment involves matching the sections of two texts with the same content in one or more languages. Alignment is defined as:

> "Sentence alignment is the problem of making explicit the relations that exist between the sentences of two texts that are known to be mutual translations." (Simard & Plamondon, 1998:59)
>
> "The problem of aligning parallel text is characterized more precisely as follows:
> INPUT. A bitext (**C**,**D**). Assume that **C** contains C passages and *C* is the set {1,2......*C*} and similarly for **D**.
> OUTPUT. A set A of pairs (*i,j*), where (*i,j*) is a *coupling* designating a passage $\mathbf{C}_i$ in one text that corresponds to a passage $\mathbf{D}_j$ in the other, and $1 \le i \le C$ and $1 \le j \le$ D. (In the general case A can indicate a many-to-many map from *C* to *D*.)" (Wu in Dale et al. (Eds), 2000:416; emphasis and typography original)

The text segments considered to be mutual alignments are called *beads.* Alignments in which beads preserve the original structure of text are known as *monotonic alignments,* that is

[1] Since 01/09/01 member of the MultiLingual Technology group at SAP AG, Germany

[2] Text alignment is used for many Natural Language Processing (NLP) tasks and applications including bilingual lexicography and terminology work, Example-based Machine Translation (EMT), multilingual Information Retrieval (IR), corpora as an information source, and word sense disambiguation.

beads occurring in the same place in both passages with no crossing over between source and target language segments.

Alignments where all the matches are of the type one-to-one are called *bijective alignments.* In bijective alignments there are no text segments left unmatched and they are therefore *total* alignments. Total alignments rarely occur in real life, other than at the highest levels of granularity e.g. document or chapter level. Most frequently in real life we come across *partial alignments,* containing some unmatched segments, known as *singletons. Many-to-many alignments* are also features of real life alignments, with one segment coupled to multiple segments e.g. {1:2, 2:1, 1:3, 3:1}. Many-to-many groupings are often caused by *crossing dependencies,* changes in the linear order of text.

Partial alignments are caused by the fact that, contrary to the underlying mathematical assumption, human translators do not always render one sentence in the source language as one sentence in the target language. The reasons for this are diverse and include syntactic, semantic and stylistic considerations. It is in the nature of localisation that not all information is relevant to all markets meaning that same text in different languages does not always contain the same semantic content.

An important consideration when investigating alignment is the notion of what constitutes an alignment. How much semantic content must overlap between a source and target language sentence pair for it to be considered a bead.

Alignment can be carried out to various document structure levels e.g. document, page, paragraph, sentence, word, etc. *Hierarchical alignment* is the approach to alignment whereby alignment is carried out at the highest granularity first before the nested constituents are aligned.

Alignment Methods

Theoretically, a variety of sentence alignment techniques[3] exist; they are based on sentence lengths, lexical constraints, and correlations or cognates.

The text type determines the triviality of the task of alignment. Much research has been carried out using parliamentary proceedings (Hansards) which have solid anchor points such as headings. As a result of consistent, literal translation they give rise to a high level of sentence and paragraph correspondence between source and target texts. Other text types can be, are however, a great deal messier, that is, contain more *noise.*

The length-based approach to alignment is more easily implemented although lexical approaches tend to give slightly better results.

The first proposals of length-based alignment techniques were put forward by Gale and Church (1991) and Brown et al. (1991) with a more thorough analysis of the results in Gale and Church (1993). Length-based methods use dynamic programming to find a *minimum cost alignment,* that is the alignment with the highest probability of being correct.

Alternatively, lexical information can be used as a guide for the alignment process, creating more robust methods of alignment which would be better able to cope with noisy imperfect input. The advantage of this approach is that it still aligns sentence beads rather than offsets, as in the previously described methods.

Kay and Röscheisen (1993) used lexical information in a computationally intensive model. Their approach is to give confirmation of alignments especially in cases of similar length, moving away from the lexically poor methods of Brown et al. and Gale and Church. They

[3] Two very thorough secondary sources (Wu in Dale et al. (Eds), 2000 and Manning & Schütze, 1999) exist dealing with this subject.

induce alignments from partial alignments of lexical elements. By using lexical cues they side step the need for prior alignment at the paragraph level.

Sentence alignment is no great problem when working with clean texts. Real life problems: less than literal translations, languages with few cognates, and languages with differing scripts pose a serious problems. In general, methods of modelling relationships are more robust and language universal. However, these techniques are still very crude in relation to fine grained document structure. The method to be used depends on the languages involved, the level of accuracy required.

Of particular interest are also the specific difficulties of aligning unrelated languages, e.g. English-Chinese, caused by the lack of structural markers, e.g. punctuation, in languages with non-Latin scripts, which complicates the alignment problem, c.f. Wu (1994).

Tools

WinAlign, the alignment tool component of the Freelance Version of Trados Translation Solution Edition 3, was used in this investigation. Using WinAlign segment pairs can be manipulated by dragging and dropping segment links to create the correct alignment. All alignments must be approved by the user before a project can be exported from WinAlign and imported into an empty translation memory in the Translator's Workbench or merged into an existing memory. WinAlign allows the user to edit text during the alignment process. The alignment results are exported in ASCII format and can be further manipulated by the user.

The first step in the process of preparing and carrying out an alignment project using Trados WinAlign is creating the alignment project. First, the settings used for the alignment are chosen, these settings have a direct effect on the way in which the alignment is carried out, for example the degree of granularity of the alignment. Next, the files to be aligned are imported into the project. After the alignment algorithm has run the results are checked, corrected and confirmed manually.

In the WinAlign user interface the alignment is displayed at several different structural levels. These correspond to the different levels of alignment which in turn correspond to the different levels of document structure in the source and target files. The WinAlign hierarchical display allows the user to first check the alignment at superstructure level (document and paragraph levels) and then at substructure level (translation units at the sentence level and lower).

Experiment

In this investigation documentation from the software company SAP AG was aligned. Software help documentation is a text type normally translated with the aid of TM, it is produced and translated in 'soft' format, has to comply with strict standards and guidelines on language style and formatting, and is updated at regular intervals (at least once or twice *a* year at SAP for each new Release).

Before alignment description and correction was carried out a framework for the evaluation and recording of data was devised. Two key factors, hierarchy of alignment and match type data, were to be investigated. They were both recorded for the raw alignment and for the corrected version of the alignment to facilitate later comparison and analysis.

It was necessary to implement a method of describing the misalignments and their knock-on effects for the hierarchy of the alignment as well as documenting the number of different match types thrown up by the alignment algorithm. Furthermore, it was necessary to record this data in parallel for both the raw and the 'hand corrected' alignment hierarchies.

The raw alignment was recorded, checked and corrected one level at a time. It was important that this process be carried out hierarchically because any misalignments at the topmost levels would mean that all segments below them were also incorrectly aligned.

First of all the alignment was checked at the file level (structure view level 1). It is possible to align up to twenty files simultaneously using WinAlign so mismatches do sometimes occur at this level. The next stage is the checking any misalignments at the paragraph level (structure view levels 2). If misalignments at this level are overlooked they will cause significant problems later. The third and most time consuming stage of checking is that of misalignments at the sentence level (structure view levels 2-5). These misalignments were recorded and their knock-on effect for the rest of the alignment also noted.

Alignments at structure view levels 1 to 3 generally include just one text segment for the source language and one text segment for the target language at both the superstructure and the substructure levels. At structure view levels 4 and 5 one match pair at the superstructure level can include many match pairs at the substructure level. These substructure beads are the aligned sentence level text segments.

The alignment hierarchy was depicted graphically at levels 1 to 5 for both the source and target text. Besides which, match type data was recorded for any substructure alignments containing more than one text segment pair. These normally occurred at structure view level 4 or 5 but in some cases they even occurred at structure view level 1. The number of text segments at this, the lowest substructure level, was recorded for both the source and the target language texts. A tally was kept of the number of substructure levels which needed correction. Match type data was recorded only for alignments at the substructure level.

The process of data collection was very time consuming and demanding, as a high degree of concentration was needed to ensure no mistakes were made.

Alignment data

Hierarchy data

The full alignment hierarchies were recorded graphically in tables. The hierarchy of the alignment is divided into several different levels, as described above. Here superstructure and substructure level are discussed separately.

As the hierarchy of the raw alignment was recorded and corrected a tally was made of the number of superstructure alignment beads which governed multi-bead substructure alignments for which corrections were necessary. Just under half (49%) of such superstructure beads required correction at the level of the substructure alignments.

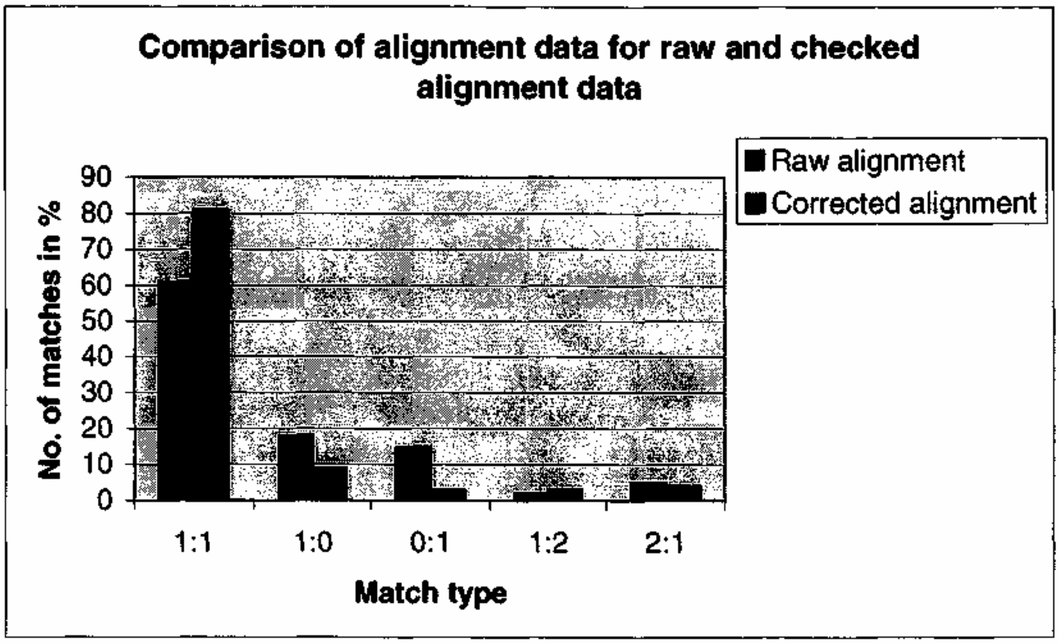
The alignment hierarchies show that 6 misalignments occurred at the superstructure level. In the results tables these misalignments are highlighted in blue (for the source language) and red (for the target language). These misalignments are important not only because they cause misalignment in all substructure segments they govern, but also because they cause a misalignment 'domino effect', causing knock-on misalignment of subsequent beads at the superstructure level.

Misalignments occur most frequently at the substructure level. Due to the fact that this type of misalignment occurs more often and is more complex, they are more difficult to characterise.

Individual examples of misalignments at the substructure level are the case in which a 2:1 misalignment must be corrected to give a 1:1 and a 1:0 match. Or the co-occurrence of a 2:1 misalignment followed by a 1:2 misalignment causing a domino effect of 1:1 misalignments until another 2:1 misalignment occurs. The original two misalignments must be corrected to three 1:1 alignments and the final 2:1 alignment also corrected in its context.

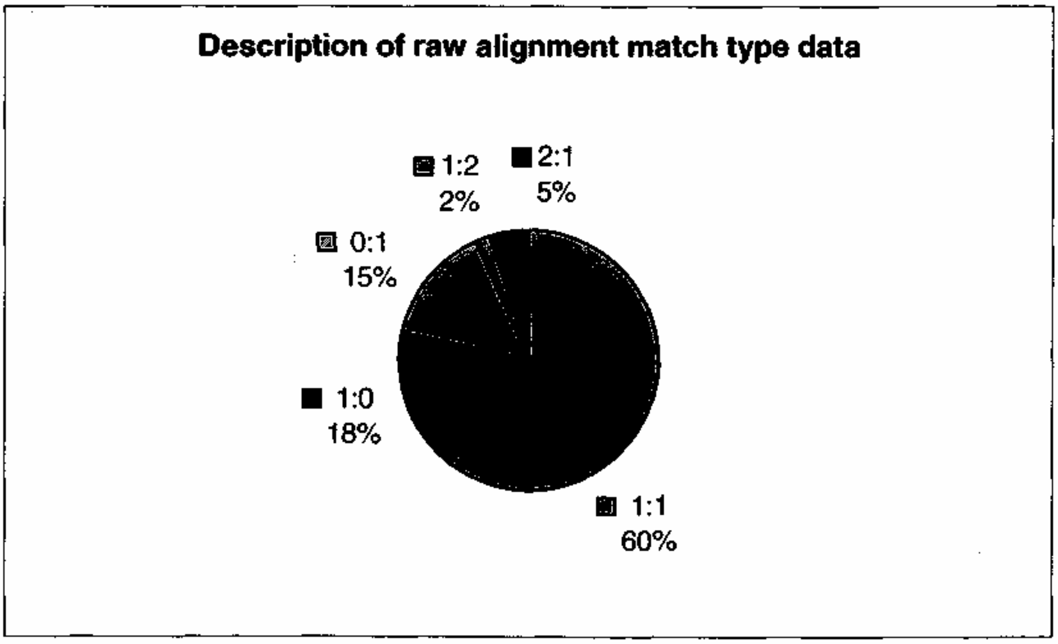
Match type data

Match type data was collected for the substructure alignment beads. This data is described in the following sections.

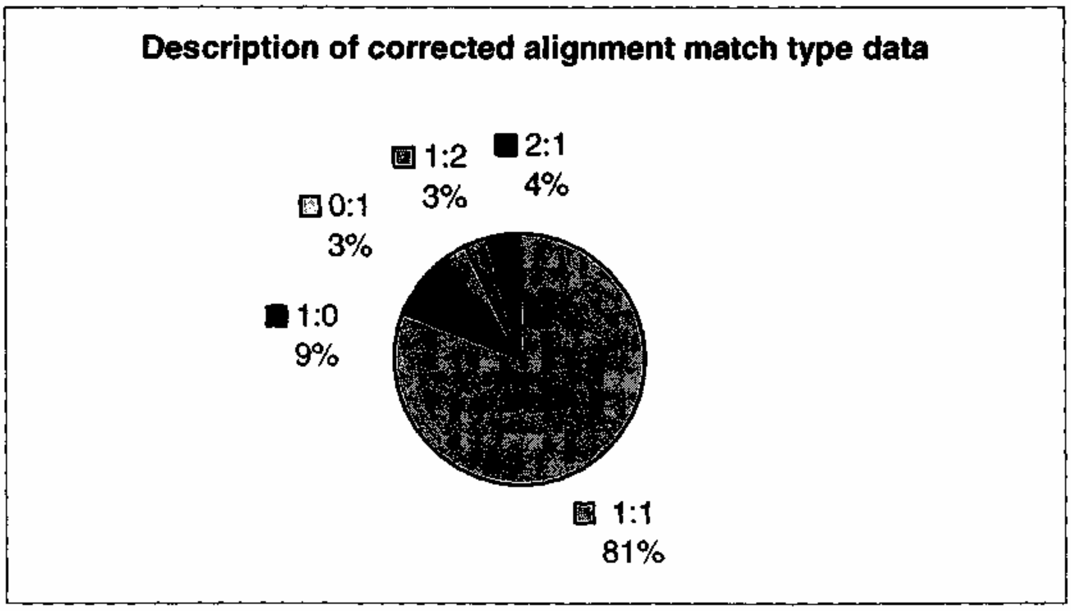


The bar chart above shows the difference between the match types which occurred in the raw and the corrected alignments. Alignments of the type 1:1 are by far the most frequent. However, the WinAlign algorithm does not always recognise them correctly. There was a tendency towards finding 0:1 and 1:0 matches which must subsequently be corrected manually. Rare alignments of the type 1:3, 3:1 and 1:4 did occur in the test corpus, but the number of such alignments was so small that they do not show up on the scale of the bar chart above. These rare alignments were not recognised correctly by WinAlign, but were found during the manual checking process. The pie charts in the following section show the breakdown of the different match types in the raw and corrected alignments in greater detail.

The pie chart below clearly shows that the most frequently occurring match type for the raw alignment is 1:1 followed by 1:0 and 0:1 type matches.



The pie chart below shows the breakdown of match types for the corrected alignment. In this case, 1:1 type matches account for a larger slice of the pie, again followed by 0:1 type matches. This time, though, 2:1 type matches make up 1% more of the pie than 0:1 type matches. In the corrected alignment 1:3 type, 3:1 type, and 1:4 type matches did occur. However, they account for a negligible proportion of the whole alignment: 1:3 type and 3:1 type matches making up 0.1% of the total each and 1:4 type matches accounting for 0.05%.



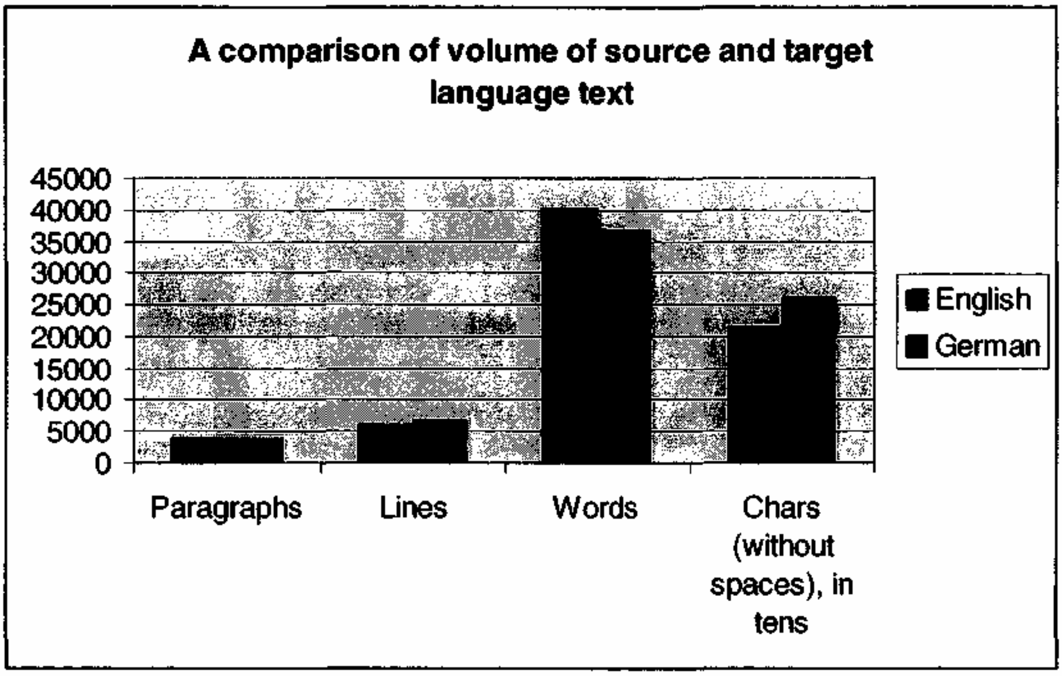
The results data show that misalignments occur at both the superstructure and substructure level of alignment. Misalignments in the alignment hierarchy can give rise to a 'domino effect' passing on false alignments down the hierarchy.

The match type alignment data shows that the largest proportion of matches are of the type 1:1, with this figure increasing further after manual checking and correction. Manual checking also finds 'rare' alignments which are missed by the algorithm.

Discussion

In this section the results of the investigation are discussed and analysed. Firstly, the implications of the volume of data studied in the investigation are looked at. Next, the results of the raw and corrected alignment, presented in the previous section, are analysed. Finally, the quality of the WinAlign alignment algorithm is discussed with reference to the factors listed above.

Volume of data



The alignment was carried out on two large pairs of documents. The bar chart above compares the volume of source and target language text in these documents. Interestingly, when measured in paragraphs, lines, or characters there is a greater volume of the German (source language) text. However when the volume of text is measured in words the quantity of words in the English (target language) text is greater.

This data shows there is 14% more text in the German version of the documentation than in the English. This value is gained when the text is measured in characters not including white space. However, when a comparison is made of document size measured in words the German text is smaller by 8%.

The fact that there are fewer words in the German documents than the English ones can be easily explained by linguistic factors. Compound nouns are formed in German by 'sticking' one or more words together, whereas English compound nouns are morphologically separate units. Thus, there are fewer and longer words in the German texts. For this reason it is sensible to take number of characters in a document (not counting white space) as a measure of volume of text as the degree of variance is likely to be a lot smaller between the two documents.

Alignment data

In the coming sections sources of misalignments are discussed. Both linguistic and stylistic influences are considered. This data will then be used in the conclusion to put forward proposals for an alignment checking tool.

Hierarchy data

At the level of alignment hierarchy data, sources of misalignment include: omission or insertion of information by the translator, differences in linguistic expression between the source and target languages, stylistic differences such as sentence length, and poor use of punctuation and formatting.

Analysing the alignment hierarchy data shows that the degree of complexity of misalignment and the causes of misalignments cannot easily be specified with any degree of accuracy.

Checking at the superstructure alignment level is particularly important as the knock-on-effects of misalignments at this level are particularly dramatic, 49% of superstructure beads governed substructure alignments which needed manual correction. These results show the necessity of a checking tool.

It is important to note that, as with the superstructure alignments, it is difficult to characterise misalignments at the substructure level and, thus, it is difficult to pinpoint the exact causes of the misalignments.

Match type data

The results of checking and correction of match type also show the need for alignment checking. The data for the corrected alignment shows that although 1:1 alignments are the most frequent, other match types do occur.

It has already been noted above, that the German prose style differs from English insofar as long sentences occur far more frequently than in English. This fact accounts for the occurrence of 2:1 type matches. The converse could be said to account for less frequent 1:2 type matches.

The occurrence of 1:0 type matches can be explained by differences in style and formatting conventions followed in English and in German. In the German text certain set phrases occurred regularly which were not rendered at all in the English, as such information is considered to be implicit in the text. There are of course cases in which the English translator may feel the need to spell out information to the English reader which would be considered implicit by the German reader, this scenario gives rise to 0:1 type matches.

Trados WinAlign

Trados' WinAlign program is based on a robust alignment algorithm. The application did not crash once during the period of investigation. The performance of the algorithm was very good in terms of speed of alignment, even the very large files which were used as a basis of this investigation aligned quickly. However, the quality of the raw alignment itself was not perfect and post-checking was essential to ensure the resulting alignment would be of use as a Translation Memory.

It is clear that Trados have had to make trade off the stability of the application, speed of alignment and modest processor requirements against the degree of alignment accuracy.

An application to aid the task of alignment checking would be of value, saving on time and manpower invested in the alignment task and improving the overall accuracy and quality of the alignment.

Conclusion

In this section conclusions are drawn from the information gained during this investigation. Firstly, suggestions are put forward as to how measures in the documentation preparation process could ease the task of alignment. Secondly, search keys are proposed for a potential alignment checking tool. Next, this tool is described in some detail. Finally, suggestions are made for further work which could be carried out following on from this project.

Document preparation

Due to the fact that SAP documentation is written to conform to a relatively strict framework of standards and guidelines it was possible to carry out document structure level checks quickly. However, some of the inconsistencies detected at the levels of document structure and formatting between the source and target text could have led to a lower quality alignment.

Rigorous checking carried out at all stages of the authoring process would prevent certain types of misalignments from occurring. The purpose of such checks would be: firstly, to ensure that the content and structure of the source and target documents are as close to one another as possible, and secondly, to ensure that no forbidden formatting which would cause problems for the WinAlign program was used in the documents.

Checks on document content and structure would entail comparing the size of the documents being checked, bearing in mind language differences. Section headings and automatically created contents tables could be used to compare the linear order of text and to check for omission or insertion of textual material.

Suggestions for keys for a potential alignment checking tool

Basically, keys for searching for potential misalignments are the misalignment patterns and their causes described and discussed above.

Match type data is one such key: 0:1 or 1:0 type matches (resulting from insertion or omission of extra textual information), many-to-one and one-to-many type matches (caused by differences in linguistic expression between languages), and so on.

Other intuitive factors such as text segment length (taking into account language variance) and segment content (cognates and constants which should occur in both the source and target language text segment) could also be used as checking keys.

Further investigation of larger aligned corpora could be used to gain statistical weightings for such key data.

When looking at different match type combinations the statistical weightings would tell the checking algorithm which matches it should 'prefer' as possible misalignments.

Tool proposal

In this section a tool is proposed for the automatic checking of alignment files. This tool would be an application integrated in a workflow process for the translation Memory creation and implementation.

The checking tool will offer the user a choice of settings which dictate which keys are used for the alignment checking. At this stage the user also gives the alignment checking tool information about the project being checked e.g. source and target languages, creation and alignment tool, etc.

Next, the alignment project exported from the alignment tool (in the case of WinAlign a plain text file) would be imported into the checking tool.

Now, the checking algorithm could be run. Following the alignment check the alignment would be displayed to the user in an intuitive format similar to that of WinAlign with high chance misalignments flagged for closer scrutiny from the checker.

The tool would offer the user several options as to what can be done with the flagged segments: ignore misalignment (if in fact the alignment is correct), correct misalignment as suggested (if a suggestion is given for correction), correct misalignment manually (if the suggested correction is considered incorrect by the checker, or if there is no suggestion).

The user has the option of rechecking or skimming the entire hierarchy to assure that the alignment is correct before saving and exporting the corrected alignment.

Further work

The goal of improving text alignment is an ambitious one. Further work is necessary before an alignment tool of the type described above can be developed.

It is necessary to invest more time into investigating and characterising crossing-dependencies and their effects. Crossing dependencies are alignments which do not occur in the same linear order in both source and target text. If the translator decides that a text reads more intuitively in the target language in a different order to that in which it has been written in the source language these changes may cause difficulties for the alignment algorithm. The phenomenon is particularly complicated. Creating a descriptive framework for this phenomenon would be aid more detailed analysis.

For statistically significant results on misalignment data this investigation should be carried out on a larger quantity of data, i.e. more documentation should be aligned and assessed.

More work is needed in describing and quantifying a larger corpus of alignment data to gain statistical data to train the alignment checking algorithm.

Summary

To summarise the conclusions of this investigation, it is suggested that more consistent authoring and translation practise would reduce the number of misalignments. To this end quality checking tools, such as term checking and the use of a controlled language, could be integrated into the documentation and translation workflow to carry out automatic checks of document structure and formatting.

A tool for automatic alignment checking was proposed as were keys which the algorithm behind this tool would use to search for potential misalignments. The keys for alignment checking include match type data, segment length, cognates and constants, and other statistically weighted data. The tool itself would be a discreet environment with an import/export function for the files to be checked. After the alignment checking algorithm had run the tool would flag potential misalignments so that the user could check them more closely.

Of course, this is a very ambitious undertaking and before the tool can be developed more groundwork must be carried out. Most important, is the collection of more alignment data so that statistically significant results can be analysed from which an algorithm could be developed and the tool trained.

References

Brown, B. F., Lai, J. C. & Mercer, R. L. (1991) *Aligning sentences in parallel corpora.* 29th Annual meeting of the Association for Computational Linguistics, Berkley, CA pp. 169-179

Dale, R., Moisl, H., & Somers H. (Eds) (2000) *Handbook of Natural Language Processing.* New York: Dekker pp. 415-459

Gale W. A. & Church, K. W. (1991) *A Program for Aligning Sentences in Bilingual Corpora.* Technical Report 94, AT&T Bell Laboratories, Statistical Research

Gale W. A. & Church, K. W. (1993) *A Program for Aligning Sentences in Bilingual Corpora.* In Computational Linguistics 19, 75-102

Kay, M. & Röscheisen, M. (1993) *Text-Translation Alignment.* Computational Linguistics 19, 121-143

Manning, C. D. & Schütze H. (1999) *Foundation of Statistical Natural Language Processing.* Cambridge, MA: MIT Press pp. 463-494

Simard, M. & Plamondon, P. (1998) *Bilingual Sentence Alignment: Balancing Robustness and Accuracy.* Machine Translation 13, 59-80

Wu, D. (1994) *Aligning a parallel English-Chinese corpus statistically with lexical criteria.* 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, NM pp. 80-87