Extending Translation Memories

Emmanuel Planas, 2000.07.31
NTT Cyber Solutions Laboratories, Japan.
[planas@soy.kecl.ntt.co.jp, planas@imag.fr]

Introduction

In this paper, we are concentrating on two important notions for Translation Memories (TM now on). The first notion is redundancy. To understand the related issue, try to answer this simple question: "What does mean <this text matches such memory at a rate of 23% of exact match>". What would match: characters of the source part of the translation unit of the memory (sTU) and the input segment ? Or perhaps the words are matching ? And what about layout attributes ? Do we take into consideration indexes, images, or other non textual data ? In order to answer these questions, we will give a precise notion of what redundancy in Section 1 of this article.

The second notion deals with text segmentation. Most translation memories are based on what are called segments. Some others on paragraphs. You can think about a segment as a sentence, but sometimes it is just a phrase or an incomplete sentence like "For more information see:". Marks play an important role in determining these segments boundaries, and most of the current market TM allows you to tune which mark you would want to be a terminator for your segments. We propose to scale down translation units (TU) at a sub-sentence level with the notion of Chunk. Some evaluation is provided here and shows how beneficial Chunks can be for TM recall.

1. Defining Clearly Redundancy

1.1. Textual or Full redundancy

1.1.a. Understanding the nature of a document

Example 1 shows one of the most common documents formats in 2000, a RTF file. In this example the meant sentence is "This sentence uses bold, italics, and combined italics and bold characters plus an index mark". The RTF coding keeps track of the layout the author has applied to this text with the edition software. "{\b }" for example indicates to the edition software (Microsoft Word) that what is between the braces has to be represented in bold. This figure illustrates that document is not a mere sequence of words, and not even of word characters, but rather a sequence of heterogeneous characters; each of these characters have to be considered as an element of the encoding system, not as part of a word: some represent words or sentence marks, others layout information, some others links or indexes for example.

The translator would interpret and manipulate this heterogeneous flow of information with the help of a word editor. He would understand what to translate, which element to transfer or adapt to the target language. The first consequence for TMs is that TMs should be able in some way to manipulate the document data and separate non linguistic data (layout marks, indexes, etc.), and at the same time keep the relation between this non linguistic data and the text itself, so as to be able to transfer it to the translated text, like the word editors allows the translator to do so.

```
....{\*\pnseclvl7\pnlcrm\pnstart1\pnin
dent720\pnhang{\pntxtb
(}{\pntxta                                    )}}
{\*\pnseclvl8\pnlcltr\pnstart1\pnindent
t720\pnhang{\pntxtb    (}{\pntxta    )}}
{\*\pnseclvl9\pnlcrm\pnstart1\pnindent
t720\pnhang{\pntxtb
(}{\pntxta    )}}\pard\plain    \widctlpar
\f4\lang1036
This sentence uses{\b bold}, {\i
italics}, and combined{\b\i italics and
bold} characters plus an index
mark{\pard\plain                \widctlpar
\v\f4\lang1036 {\xe {\index mark}}}.
\par }....
```

Figure 1: An example of RTF document

1.1.b. A suitable internal structure for Translation Memories

A first problem that this observation arises is how to represent all this data in a convenient way in Translation Memories. We have proposed to use a structure of linked lattices that we call TELA in [Planas 1998] and [Planas 1999]. We invite the readers to refer to these documents for more information, since we are not going to discuss this problem here. Just bear in mind that the TELA structure allows the TM system to separate the document data according to its nature, while keeping the link between the different elements of the document. Next Figure illustrates this:

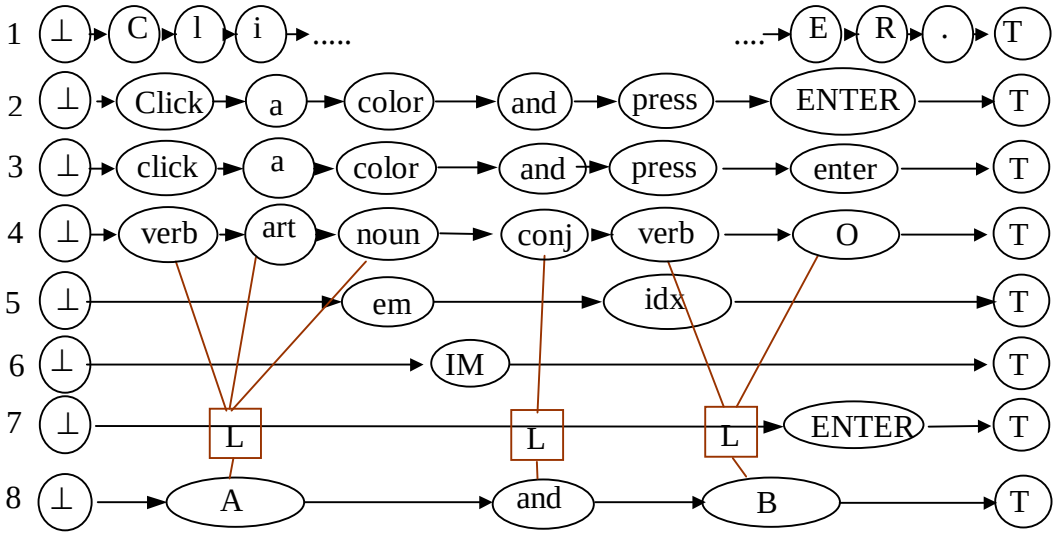


Figure 2: an example of TELA structure

1.1.c. Consequences for the Redundancy

The heterogeneity of the data included in electronic documents rises some difficulties when wanting to match such an input sentence with some sTU of the memory. We clearly see that, according to the kind of job the translator is doing, he would want to consider or not such part of the layout information or not such as indexes, images, used fonts or revision marks.

We propose to link the redundancy concept to the nature of the document data that is taken into consideration for the comparison of segments. The data to be considered could for example simply be the text itself. In this case layout information and other non textual

document data would not be consider. The following two RTF-like represented sentences would then be considered as equal:

Click a {\b color} and press {\idx ENTER}

Click a {\b color} and press ENTER

Example 1: two RTF-like formatted sentences to be compared

This is easily achieved with the TELA representation of these sentences: it suffices to compare only layer 2 bearing only the words of this sentences. More importantly, the two following sentences, one represented in a RTF format and the other in a HTML format would also be considered as equal:

Click a {\b color} and press ENTER

Click a color and press ENTER

Example 2: HTML-like and RTF-like formatted sentences to be compared

This kind of redundancy allows of course a memory built in one format to be used by a different format. We call it "**textual redundancy**".

As opposed to textual redundancy, we propose to consider "**full redundancy**" as the one taking also into consideration non textual data. In terms of "full redundancy", the sentences of Example 1 are not equal. Please note well that, if the internal representation use by the TM is correctly designed, Example 2 sentences should be considered equal in terms of full redundancy also.

Moreover the non textual elements to be taken into consideration for the full redundancy should be adaptable. One could be willing to take into consideration or not indexes, revision marks, bold but not italics, etc.. The TELA structure will authorize this if one layer is attributed to each non-textual data to be taken into consideration.

As a conclusion for this first approach to redundancy, we can say that according to the elements of the electronic document we are considering, the nature of redundancy takes different meaning. We propose to separate clearly the **textual redundancy** where only the meant text is taken into consideration, and the **full redundancy** where the meant text plus all or part of the non textual elements are taken into consideration.

1.2. Text units

In a TM system, the text is cut down into text units. Those text units will become the source part sTU of the translation units TU. The target part tTU being the translation of sTU. The usual way to do this is to separate the text in segments according to predefined **non linguistic** rules. Upper case letters and marks play a major role in this segmentation, and this results in getting roughly sentences as text units (that we call segments because of the "roughly"). When trying to find a corresponding TU in the memory, the input current segment is compared to sTU which are also segments. The redundancy is then generally based on segments (sentences).

The segmentation of the text could well be defined more globally by paragraphs like in some internal company tools (this solves the difficult problem of text segmentation), or more precisely by looking at sub-sentence parts like phrases (see [Gaussier 1999]) or chunks like we are going to propose later (see Section 2 of this article). Taking the word or the character as text unit is the extreme position of this logic, in this case the translation memory is called a terminology database and has to be entered by hand. The automation of the collection of the source and the target words being somehow difficult, this option has to be dropped as far as TM are considered. I order to catch the importance of text units in terms of redundancy, let us consider the following text, separated into sentences (periods) and chunks (brackets):

"[When encountering] [a system error], [refer first] [to the manual] [or to the online help]. [A system error] [can sometimes be repaired] [by killing] [the current process]. [To kill] [the current process], [right click] [the mouse] [when pointing] [the lower bar], [and select] [task manager]. [Select] [the current process] [right click] [the mouse] [and select] ["delete"].

[When encountering] [a warning], [refer first] [to the manual] [or to the online help]….."

There are 5 sentences in this text. The first and the last sentence are redundant (if non 100% match are accepted). This makes 40% (2/5) sentence redundancy.

There are 28 chunks. The first and last five chunks included in the first and last sentence are of course redundant. This makes 10 redundant chunks. In addition:

[a system error] appears in the first and second sentences: 1 more redundant chunk

[the current process] appears in sentences 2, 3 and 4: 3 more redundant chunks

[right click] [the mouse] [and select] appear twice: 6 more redundant chunks

In a whole this makes 20 redundant chunks, say 71% of chunk redundancy.

The choice of the text unit is then essential to the definition of redundancy, and one should clearly tell which text unit he chooses for expressing the redundancy he is talking about.

1.3. Inter and intra redundancy

The translator's job, or the translation agency job not only has to cope with the text itself, but it has also to take into consideration how to manage files. When you have a set of documents to be translated (say a "job"), you know you are going to be helped by a Translation Memory system if you are in one of these two positions:

either you have the memory of a previous job applicable to the current job

or the current job contains repetitions

Those are two separate forms of redundancy. We call the first one "inter redundancy", and the second one "intra redundancy". We have compiled the following evaluation for you to well appreciate the difference.

Let us consider a document composed of one hundred sentences. Let us also suppose that the sentences of this document can be partitioned into three parts:

Part 1: 30% of the sentences can be found in the memory of a previous job (30% of inter redundancy), and appear only once in this document (no intra redundancy)

Part 2: 30% of the sentences appear twice in this document (intra redundancy), but can not be found in a memory (no inter redundancy)

Part 3: 40% are brand new sentences and appear only once.

Parts 1 gives 30% of inter redundancy. These sentences do exist in a previous memory, and the TM system will give them for free to the translator: 30% of the job is saved.

Parts 2 give birth to 30% of intra redundancy. But this time no memory exists. So the translator has to translate each of these sentences once. Provided that the TM system does record every translation online, the similar sentence (remember each sentence of Part 2 appears twice) will be provided by the TM system. So in a whole, the translator will have being translating 15% of the sentences, and will get for free the other 15%.

This illustrates that **inter redundancy** and **intra redundancy** are then two different notions that TM systems should clearly separate. Note that the previous example was simplified though. In fact the link between intra redundancy and productivity gains depends on the frequency of the redundant sentences, and the mean number of words each sentence contains: the more frequently reused are the sentences, and the more numerous are the words in the repeated sentences, best is the benefit. See Chapter 1 of [Planas 1998] for more information about this point.

1.4. Summarising the different criteria for text redundancy

We have seen three criteria for qualifying redundancy:

Textual and Full redundancy

Text units: segment or chunks for example

Inter and intra redundancy

Other criteria could be taken into consideration. Further more, the granularity for each criteria could be different according to the needs: we have see for example that non-textual data could be differentiated into more precise sub-categories. In order to settle this notion in the lector's mind we propose this simplified schemata where only three criteria are proposed and each criterium is bi-valued. This gives eight (2^3) kinds of redundancies:

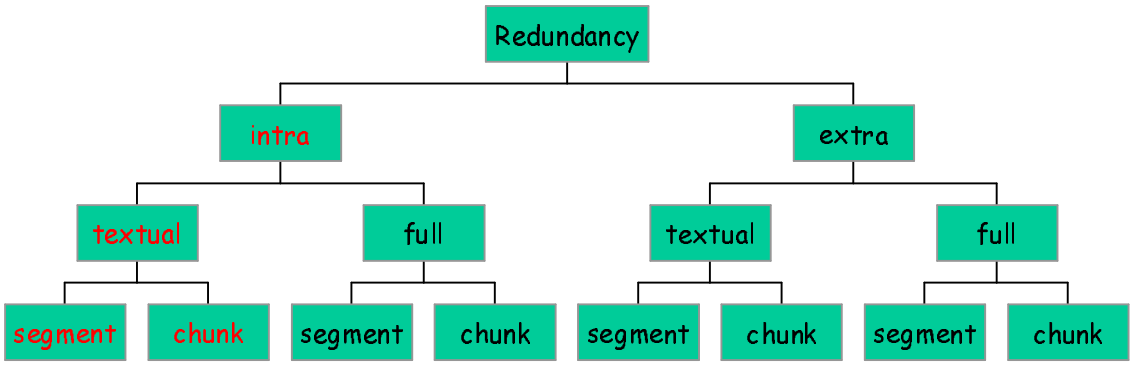


Figure 4: A possible differentiation of redundancies

1.5. Measuring Redundancy

1.5.1. Basic method

A kind of redundancy and a document being chosen, the redundancy against a given memory is the number of sentences of the document that are also in the memory. If the document is too large, we can measure it by extracting a representative test set (random extraction for example) of text units from this document and search in the memory which proportion is matched by some corresponding text unit of the memory. An extrapolation will give the redundancy of the text.

1.5.3. Similarity Threshold rather then "fuzzy" and "exact" match

When checking if such a segment of the document matches a TU of the memory, one can apply a boolean approach: all of the data of the segment (according to the redundancy chosen) matches or not a TU. Or one could consider that a TU matches "somehow" ("fuzzyly" according to current market tools) if a part only matches. We have proposed in

[Planas 1999] to clarify the "fuzzyness" using a proper defined similarity. This similarity is a multi-layered similarity that measures the proportion of items matched at each layer of the TELA structure. For a textual redundancy for example, if surface words, basic words and parts of speech are the three layers chosen for measuring the similarity, the following sentences have a similarity of (4/6, 5/6, 1) because four surface words out of six are matching (except for "Click" and Color"), five basic words (except "color"), and all part of speech do match.

Click a color and press ENTER.

Clicks a button and press ENTER.

Choosing a layer of data as reference, like for example basic words, and a given threshold for the this data similarity (75% for example), one gets a formal definition of the "degree" ("fuzzyness") of the match between the sentence and the sTU. For more information, please refer to [Planas-2000]. In the evaluations further shown in this article use the basic words, all TU matching the input sentence with a lemma similarity superior to the threshold are accepted. For non textual redundancy, non-textual data could be used in a similar way to get the expected result.

2. Chunks for extending Translation Memories

2.1. Definition

2.1.a. Basic notions

The notion of chunk has been introduced by Steven Abney [Abney 1991, 1996]. A current European Project called "Sparkle" [Sparkle 2000] deals with it, and some psychological evidences [Juola 1995] as well as research in speech repair [Nakatani & Hirschberg 1993] seems to give it some scientific foundation.

The author is not the first one to use chunks for Natural Language Processing applications. The reader can refer to [Zechner & Waibel 1998] who use a chunk parser to analyze speech input, or [Veale & Way 1997] who indeed used a text segmentation similar to chunks, without explicitely referring to it and applied it to Example-Based Machine Translation.

The approach to chunks that is going to be presented here is not theoretical. It is intuition and data guided, and should require more linguistics studies like the ones cited above. Let us first give some examples of **chunks**. The following text is segmented into chunks represented in brackets:

[A chunk] [is] [a sub-sentence part]. [A chunk] [is not always] [a phrase] [because the segmentation] [of a sentence] [into chunks] [does not use] [the logical structure] [of the sentence]. [In fact] [it] [only uses] [the value] [of the word categories]. [This] [allows chunking] [segments of text] [not necessarily well formed] [into a sentence].

Figure 5: Segmenting an English text into chunks

There are two important notions to use for chunking a text:

Separators: separators begin or stop a chunk

Friend categories: friend categories cluster together to form chunks.

A chunk is then a contiguous group of words belonging to the same friend category, and whose boundaries are either a separator or a word belonging to a different friend category. The two notions seem to be universal, but the set of separators and friend categories depend on the language to be chunked.

2.1.b. The case of English chunks

Here are the list of separators and friend categories we propose for the English language:

Separators

Articles: a. the, …

Other function words: at, in, while, when, and,…

Friend categories

Separator word category: all separators

Verb category: verbs and adverbs

Noun category: articles, nouns, adjectives and adverbs

As there is an ambiguity for the binding of the adverb, we let the verb get the priority against the adjective in case of duality. Separators gather together to for a special case of friend category, stop the left side chunk and are part of the right side chunk. Applying these simple rules, one should be able to cut any text into chunks. The previous segmented text illustrates these rules.

2.1.c. The case of Japanese chunks

The Japanese language is particularly set for allowing to separated the text into chunks. This mainly comes from the fact that it exist special particles called "joshi" that constitutes ideal separators. Here are the separators and friend categories proposed for the Japanese language:

Separators

Joshis: "ni", "no", "wo", "ga", ..

Ending of verbs: "tara", "nara", "nai", "te", "ru"..

Friend categories

Separator word category: all separators

Non separator category: all non separators

Here is a transcribed example of Japanese sentence segmented into chunks. The separators are underlined.

[sekkai hajime no] [kukiatsu ni yoru] [idou shiki kamera wo] [secchi]

[earth first time] [hydraulic by] [movement enabled camera] [establish]

This is the first time on earth that such an hydraulic powered moving camera is set-up.

In Japanese, separators end the left side chunk and belong to it, as opposed to English where it belongs to the right side chunk.

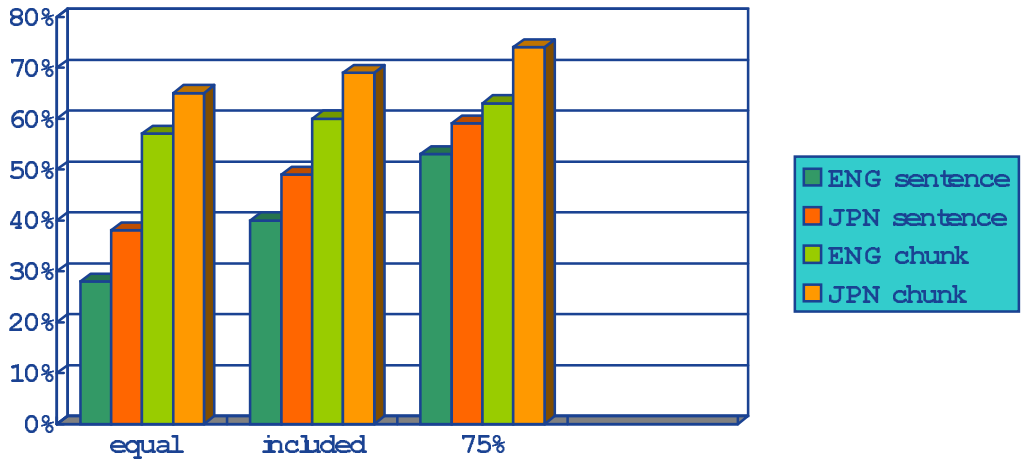
2.2. Chunks and redundancy: first clues of the usefulness of chunks

It seemed interesting to evaluate the redundancy of chunks and compare it to the redundancy of segments (sentences). To do so, we took three very different corpora, and we constituted three separate memories of segments and a memories of chunks. We randomly extracted from each corpus several test sets of segments and used these segments and the chunks included in the segments to evaluate the intra textual segment and intra textual chunk redundancies. Here are the results.

2.2.a. First corpus: tourist information

This aligned corpus consists in sentences extracted from tourist information texts like restaurants descriptions, transportation and monument guides. The corpus has been created

by NTT Cyber Solution Laboratories to be redundant. The corpus exits in Japanese, English, Spanish, Korean, Chinese and French: we used the first two languages. There are 8000 sentences that represent around 40000 chunks in both languages the exact number of chunks is slightly different in Japanese and English).

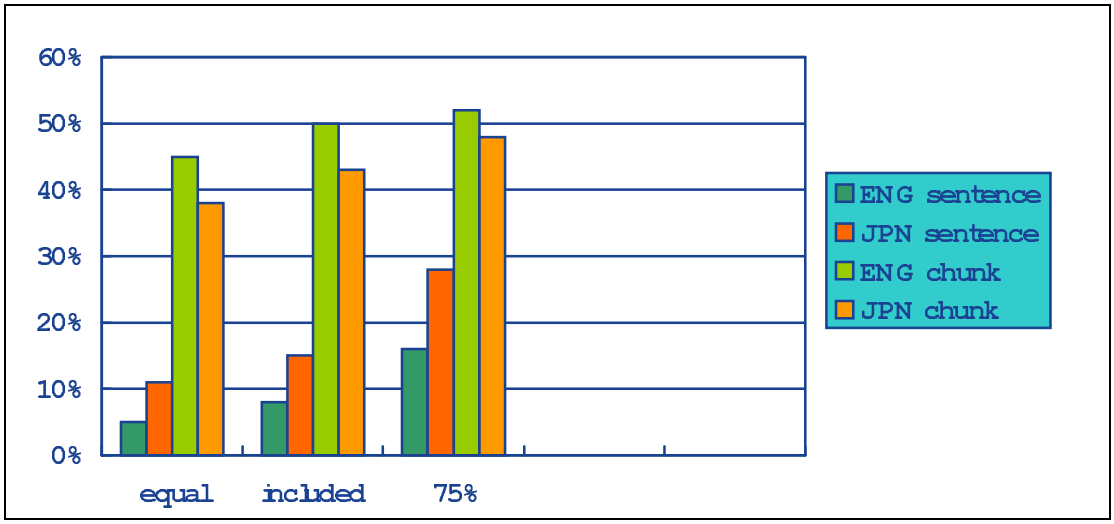


The graphic represents the percentage (vertical axis) of the sentences (1st and 2nd bars) or chunks (3rd and 4th bards) of the test set that have been matched in the memory: therefore it represents the intra textual sentence and chunk redundancy. The set of the first four bars ("equal") give these redundancies with a similarity of 100%: if a similar sentence or chunk is found in the memory, it is exactly the same as the test one. The second set ("included") gives the proportion of test sentences or chunks that are completely included in one TU of the memory. The last set ("75%") indicates the TUs found have a basic word based similarity of 75% with the test sentence or chunk. It clearly appears from this graphic that, the Chunk redundancy is drastically superior to the Sentence redundancy, for the English as well as for the Japanese language.

2.2.b. Second corpus: Software help files

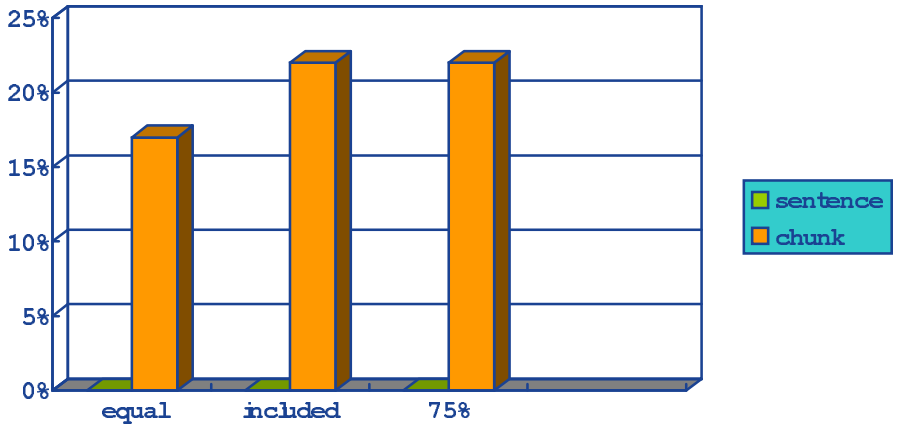
This corpus comes from software help files. Sentences have been isolated. The corpus is given in Japanese and English. And we have tested both languages. There are 8000 segments that given 39000 chunks in English and Japanese. The significance of the graphic is the same as for the previous corpus.

One can see here that the difference between the sentence and the chunk redundancy is even more important than for the previous corpus: the chunks are here 4 times more redundant than the sentences, for both languages.



2.2.c. Third corpus: Parliament discourse: the Hansard corpus

The current corpus is part of the English side of the Hansard corpus. We have taken 5000 sentences representing 50000 chunks. This experiment gives perhaps the most exciting result: None of the randomly chosen sentences from the test set (10 different test sets have been chosen) can find a correpondant TU in the first 5000 sentences of the Hansard. Yet around 20% of the chunk do find a similar one !!

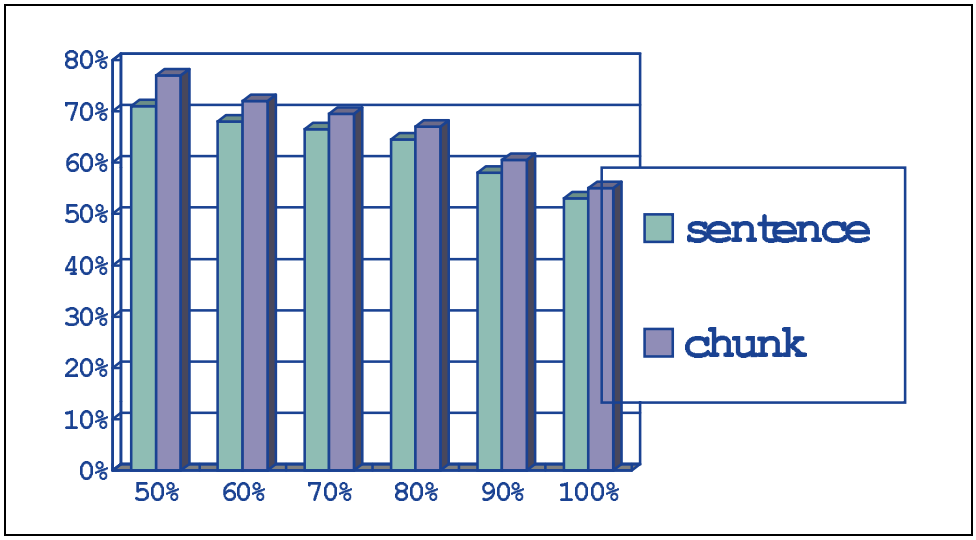


2.3. A coverage of TU chunks rather than a unique TU

The chunk redundancy being always higher than the sentence one, we wondered if, instead of looking for only one similar sTU for matching the input segment, we could not rather look for a sequence of chunks extracted from different sTU in order to match each chunk of the input. This would give the translator the optimal TU in case there is one, or a sequence of chunks covering the chunks of the input in case there are no satisfactory TU in the memory.

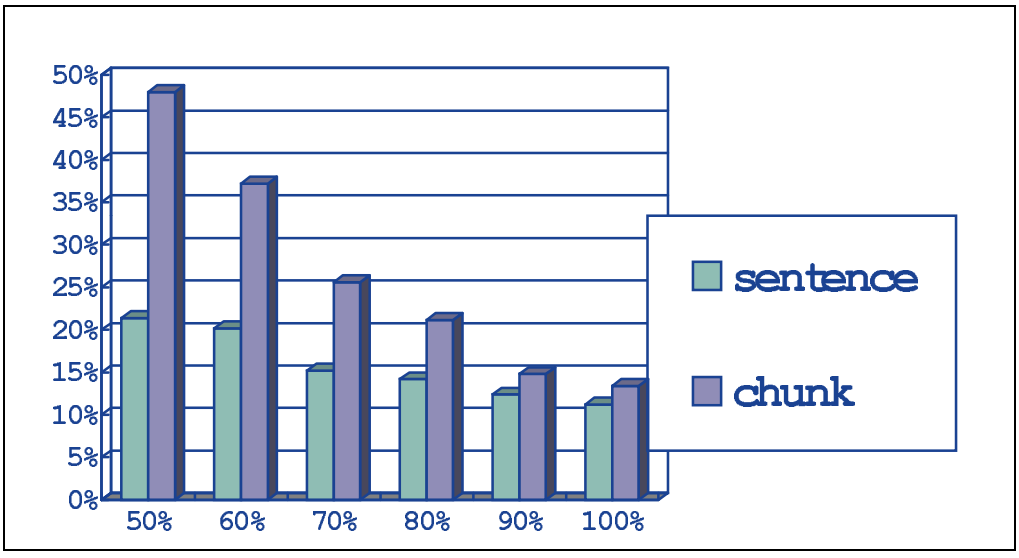
In order to test this idea, we used the three previously presented corpora, and try for each of them to cover several randomly chosen test sets of segments either by one TU or by a coverage of chunks. Here are the results.

2.3.a. First corpus: tourist information



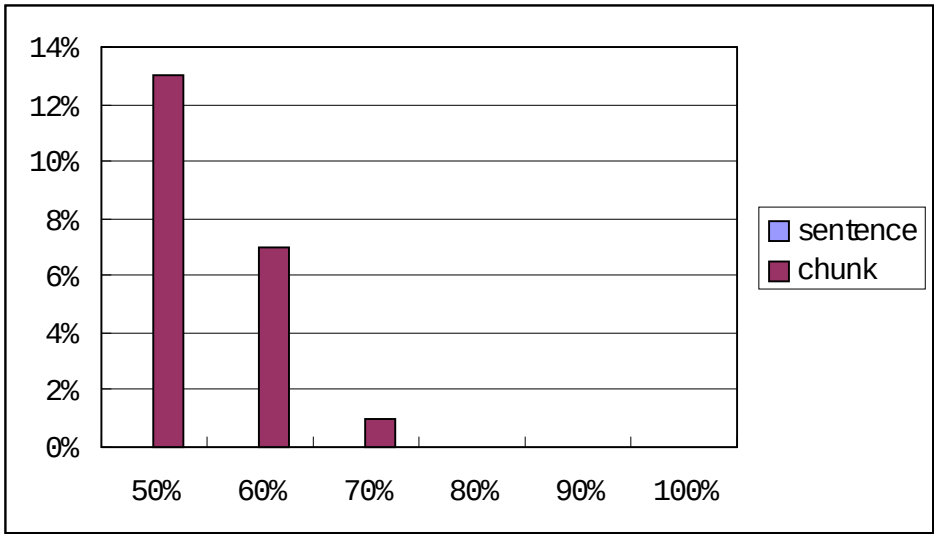
The vertical axis represents the percentage of test sentences or chunks matched by on TU segment of chunk. The horizontal axis give the result for each similarity level: 100% represents a TU matching exactly, 70% represents a match between a TU segment or a set of TU chunks and the input segment where for example 7 basic words out of 10 would be equal. Here, coverage of chunks give slightly best results than a unique segment match. In fact the test sets are already very well covered by sentences and it is difficult to get better results !! But still the coverage is better.

2.3.b. Second corpus: Software help files



The advantage of a coverage of chunk against a unique TU segment appears clearly here. Using chunks, the translator is given more help.

2.3.c. Third corpus: Parliament discourse: the Hansard corpus



Finally, with only 5000 English sentences of the Hansard corpus in the memory, we are beginning to get a coverage of chunks whose match is at a similarity greater than 50%, where no match is possible with only sentences.

2.3.d. Summarising

For all the three very different corpora, looking for a coverage of chunks instead of a match by a unique sentence gives a best recall. The recall seems not to be liked to the differential between sentence and chunk redundancy of the text, but rather to the global average of sentence redundancy: the slowest is the sentence redundancy, better the benefits of chunks will be.

Conclusions and Perspectives

Conclusions

We have proposed a definition of redundancy and tools to evaluate it. We specially believe that the redundancy and the retrieval procedure better apprehended if these kind of information is available:

- Linguistic data
- Multi-level similarity
- Change Segments for Chunks after changing linear XML for structured TELA

Our next important claim for this topic is first that redundancy (and the associated retrieval procedure) has to be characterized for example by means of:

- Extra or intra document redundancy
- Textual or full (non-textual content also) redundancy
- Text unit (chunks, sentences, paragraphs)

As far as chunks are concerned, we have showed that they give a better redundancy than sentences, and that they allow TM to get a better recall. Experiments should be conducted with a large number of sentences to see towards which coverage we can go but this first result is indeed encouraging .

Perspectives

As a natural feeling, it seems that trying to use smaller parts of sentences rather then full sentences to cover the input allows for a better recall. The important point that this study proved is that the sequence of matching chunks does better cover the input sentence than a single sentence from the memory. Other research in Statistics-Based MT like [Brown et al. 1993] or for example more recently [Langlais et al. 2000] , as well as in Example-Based MT like [Brown 1996], [Veale & Way 1997] or [Cranias et al. 1997] take benefit of this approach. These works use different kind of sub-sentence partitioning techniques. The next question to be answered for translation memory is: what kind of sub-sentence partition gives best results: grammatical clauses, chunks or n-grams. We hope to be able to conduct some experiment to give some elements to this question. Similar models helped aligning source and target sentences. The same question arises then for the efficiency of chunks related to n-grams for the alignment purpose.

The experiments presented here did indeed show that using chunks helps TM getting a better recall. We did not evaluate the precision or usability of these covering sequences of chunks. This should be done in a next step.

We would also want to complete our model and conduct a covering range of experiments for the use of chunks in a EBMT framework. Composition and adaptation within this context should give equivalent results to those of [Colings & Cunningham 1997] and [Andy and Veale 1997] so we would want to test this points.

We believe that this approach can also help the transfer of non linguistic data from the source to the target sentences in TM or EBMT. As our TELA structure allows to keep this kind of information, we are also targeting some experiments to prove such assertion.

References

Abney S. (1991) *Parsing by Chunks*, in *Principle-Based Parsing*, edited by Berwick R., Abney S. & Tenny C., Kluwer Academic Publishers 1991.

Abney (1996) *Chunk Stylebook* (work in progress) http://sfs.nphil.uni-tuebingen.de/~abney/Papers.html#96i, University of Tuebingen, Germany.

Brown F.P., Della Pietra S.A., Della Pietra V.J., Mercer R.L. (1993) *The Mathematics of Statistical Machine Translation: Parameter Estimation*, Journal of Computational Linguistics, Vol 19, #2.

Brown D.R. (1996) *Example-Based Machine Translation in the Pangloss System*, COLING-1996, Copenhagen, Denmark, pp. 169-174.

Collins, B., Cunningham, P. (1997). *Adaptation Guided Retrieval: Approaching EBMT with Caution*. 7th International Conference on Theoretical and Methodological issues in MT, pp. 119-126.

Cranias, L., Papageorgiou, H., Piperidis, S. (1997). *Example retrieval from a Translation Memory*. Natural Language Ingineering 3(4), Cambridge University Press, pp. 255-277.

Gaussier E., Hull D. & Ait-Mokhtar S. (1999) Term Alignment in Use: Machine-Aided Human Translation, http://www.xrce.xerox.com/publis/mltt/mlttart.html

Juola P. (1995) Learning to translate: a psycholinguistic approach to the induction of grammar and transfer functions, Doctor Thesis of the University of Colorado, Boulder, USA

Nakatani C. & Hirschberg J. (1993) *A Speech-first model for Repair Detection and Correction*, 31st Annual Meeting of the ACL, Columbus, Ohio, USA, pp. 46-53.

Langlais P., Foster G. & Lapalme G. (2000) *Unit Completion for a Computer-aided Translation Typing System*, 6th ANLP conference, Seattle, USA, pp. 135-141

Planas, E. (1998). TELA: *Structures and Algorithms for Memory-Based Machine Translation*. Ph.D. thesis, University Joseph Fourier, Grenoble, France (in French).

Planas, E. & Furuse O. (1999). *Formalizing Translation Memories.* Machine Translation Summit VII, Singapore, pp. 331-339

Planas, E. & Furuse O. (2000) Multi-level Similar Segment Matching Algorithm for Translation Memories and Example-Based Machine Translation, COLING-2000, Saarbruecken, Germany (to be published).

Sparkle project (2000) http://www.ilc.pi.cnr.it/sparkle/sparkle.html,

Veale, T., Way, A. (1997). *Gaijin; A bootstrapping, Template-Driven Approach to Example-Based MT*. http://www.compapp/dcu.ie/tonyv/papers/gaijin.html/, Dublin City University, Ireland.

Zechner K. & Waibel A. (1998) Using Chunk Based Partial Parsing of Spontaneous Speech in Unrestricted Domains for Reducing Word Error Rate in Speech Recognition, COLING-1998, Montreal, Canada, pp. 1453-1459.