

Identifying Fixed Expressions:

A Comparison of SDL MultiTerm Extract and Déjà Vu's Lexicon

María Fernández-Parra & Pius ten Hacken, Swansea University

116435@swansea.ac.uk; p.ten-hacken@swansea.ac.uk

1. Introduction

The term *fixed expression* refers to a formally quite heterogeneous group of expressions, such as adjective-noun collocations (e.g. *heavy smoker*), prepositional expressions (e.g. *in spite of*), verbal expressions (e.g. *break the ice*), dual expressions (e.g. *black and white*), foreign phrases (e.g. *per capita*), etc. The properties that unite them are that they consist of more than one word and are stored in the mental lexicon as one unit because their meaning is not always fully compositional.

Fixed expressions are crucial because a large part of what we say and write is made up of such expressions, rather than separate words (Mel'čuk 2001: 24). Furthermore, despite their pervasiveness in all styles of language use (cf. Pulman 1993: 250) and despite the fact that they are one of the main problems in translation (cf. Coulmas 1979: 255) and in natural language processing (cf. Sag et al 2002), they do not play a significant role in the design of CAT tools and there are no dedicated functionalities for identifying them. However, by using CAT tools to create glossaries of fixed expressions, we might overcome to some extent two difficulties in the treatment of such expressions.

On one hand, fixed expressions can be extremely transparent, such as *if and when* and *from the top down*, but they still have to be recognized as fixed. As we found by the analysis of a bilingual translation corpus, non-recognition occasionally leads to rather infelicitous mistakes in translation. On the other hand, many fixed expressions (e.g. *time and effort*, *in process*, *over time*) cannot be found in dictionaries and some more recently established expressions, such as *up line support*, are even less likely to be included in dictionaries (Colson

2004: 36). In this particular case, *up line support* refers to staff receiving help from managers higher up in the organization hierarchy.

Therefore, in this paper we explore the contribution that CAT tools can make in practice to the treatment of fixed expressions. As a starting point, we included SDL Trados (Studio 2009 version) in our experiments, as the current leader in the CAT tools market. In particular, we focused on one of its functionalities, SDL MultiTerm Extract, which we analyzed in terms of its potential as a tool to automatically identify fixed expressions from a source text. The expressions identified by MultiTerm Extract can be subsequently stored in another functionality of SDL Trados, the SDL MultiTerm termbase.

Because the object of our research is fixed expressions, rather than terms, it would be conceptually attractive to store terms and fixed expressions in different types of database. This is possible in Atril's Déjà Vu (Déjà Vu X version), as this CAT tool offers not only common tools in CAT packages such as the terminological database (termbase) and the translation memory (TM), but also the Lexicon as an additional database. This is why we included the Lexicon in our experiments. The aim of this paper is to evaluate and compare how the treatment of fixed expressions would be implemented in practice in two different CAT tools.

1.1. MultiTerm Extract vs. the Lexicon

MultiTerm Extract is designed to extract candidate terms on the basis of the statistical analysis of the input text and a list of stopwords. By contrast, the extraction methods in the Lexicon are closer to those of a concordance tool in that all words and expressions in a text are listed, without making any selection. Whereas MultiTerm Extract uses a hidden algorithm and aims to propose terms that are relevant to a particular domain, the Lexicon generates a straightforward list based on the analysis of a particular document (or set of documents). MultiTerm Extract is further discussed in section 2 and the Lexicon in section 3.

Although both MultiTerm Extract and the Lexicon can extract potentially useful strings from the source text, they also differ in the method of extraction and the scope of use for the extracted strings. In this paper we compare their relative merits in the task of identifying fixed expressions.

1.2. *Term extraction vs. Term recognition*

From the tasks that CAT tools can perform, term extraction and term recognition are relevant from the viewpoint of fixed expressions. They are fundamentally different tasks, although there is some confusion in the literature whereby sometimes term extraction is referred to as term recognition (e.g. Bowker 2002: 82).

In this research we will refer to *term extraction* as the operation of finding new items for their inclusion in a termbase, typically prior to translation, whereas we will refer to *term recognition* as the operation of matching entries from a termbase to segments in the source text, typically during translation. For the sake of convenience, we group recognition and extraction as co-hyponyms under the term *identification*.

Both SDL Trados and Déjà Vu can perform term extraction and term recognition. In SDL Trados, MultiTerm Extract is the dedicated tool for term extraction, whereas term recognition takes place in the background and the results are displayed on a pane within the working interface of the program. Déjà Vu does not have a dedicated term extraction tool. However, the Lexicon has some of the features of term extraction tools, as will be discussed in section 3 below. As in SDL Trados, term recognition in Déjà Vu also takes place in the background and the results are displayed on a separate pane within the working interface. Term recognition can also be in batch mode if using a pre-translate feature to batch translate a document.

The term extraction process in MultiTerm Extract may be viewed as a ‘black box’ in the sense that we do not have access to the exact inner workings of the program and yet it is clear that some sophisticated procedures are in place based on the selective results returned. By contrast, in the Lexicon the procedure is entirely transparent, in the sense that every word and phrase from the source text is extracted and returned in the output.

In this paper, we concentrate on the automatic extraction capabilities of MultiTerm Extract and the Lexicon. The automatic recognition capabilities of SDL Trados and Déjà Vu fall outside the scope of this research.

2. Extracting fixed expressions with MultiTerm Extract

The typical use of MultiTerm Extract as an extraction tool is to speed up the process of creating terminological glossaries prior to translation so that, when translating a document, the translation of the terms remains consistent throughout. MultiTerm Extract uses statistical methods to extract candidate terms from a text by extracting strings on the basis of the statistical analysis of the input text and a list of stopwords (cf. section 2.1 below). In the following sections we describe the settings and post-editing tools of MultiTerm Extract within a typical term extraction workflow, which we apply to the extraction of fixed expressions.

2.1. Settings in MultiTerm Extract

Term extraction with MultiTerm Extract can be monolingual or bilingual and the relevant settings for fixed expressions include minimum and maximum term length, silence/noise ratio and stopwords. For a more detailed description of how the different settings available in MultiTerm Extract influence the contents of the output when using MultiTerm Extract to extract fixed expressions, cf. Fernández-Parra & ten Hacken (2008). Here we provide a brief overview.

The minimum and maximum term length settings refer to how many words a returned string should have. Since we defined fixed expressions as consisting of two words or more, the minimum term length is set at 2. The maximum term length must be set taking into account that the returned strings often contain a certain amount of intervening material, because the component words of fixed expressions do not always occur adjacently or with the same word order.

For example, in order to identify the expression *make a contribution*, which in our source text occurs in the segment *Funding is only one component of the significant contribution the public sector makes to HIV vaccine and microbicide research*, the maximum term length needs to be set at 5 at least, so that the extracted string contains every word from *contribution* to *makes*. This is because with MultiTerm Extract it is not possible to extract *contribution* and *makes* alone in one segment. Maximum term length 4 might yield either the segment *contribution the public sector* or the segment *the public sector makes*.

The silence/noise ratio is a scale of 11 points ranging from the maximum noise to the maximum silence. Noise means that the program has extracted unwanted items, whereas Silence means that the target items have not been extracted. The higher level of noise selected, the higher the recall will be, but at the expense of larger amounts of unwanted items. Similarly, the higher level of Silence selected, the fewer unwanted items will be returned, but at the expense of not returning some of the target items.

For the sake of convenience we will refer to the maximum noise level as 1 and to the maximum silence as 0, with 0.1 increments for the levels in between. However, in previous research (Fernández-Parra & ten Hacken 2008) we verified that in fact there is no difference in output between noise levels 0.8, 0.9 and 1, which we now group as *High* noise levels. Similarly, we discovered that there is no difference in output between noise levels 0.6 and 0.7, which we now group as *Medium* noise levels. This reduces our list of noise levels to 8 different ones, from 0 to 0.5, Medium and High.

Stopwords are words to be excluded from the output. MultiTerm Extract provides two types of default stopword lists, a file called *Stopwords* and a file called *Basic Vocabulary*. In MultiTerm Extract, the stopword list and the basic vocabulary list are grouped under the term *Exclusion files*, but they are specified at different stages in the project setup process.

Both stopword and basic vocabulary lists are language-specific. MultiTerm Extract provides a default stopword list for fourteen different languages but the default basic vocabulary list is only available in five languages. The stopword list contains 392 function words such as prepositions, pronouns, verb contractions, etc., for example *on, off, my, yourself, won't, mustn't*. The basic vocabulary list contains 4,279 single words from all categories, such as *Africa, thought, ventilation, your, yesterday*. There is considerable overlap between the two lists, but they are customizable. Alternatively, users can create their own stopword lists, which can be used together with the default lists.

Since many fixed expressions contain function words, for example, *all but, out of pocket, if and when*, one would expect a priori that the use of stopword lists and basic vocabulary lists would give worse results in our experiments. However, we discovered that using both of MultiTerm Extract's default lists

reduces noise considerably without excluding many fixed expressions from the output. We discovered that the best compromise between recall and precision was obtained by using both lists. Therefore, we included both the default basic vocabulary list and the default stopword list in all our experiments.

2.2. *Post-editing tools in MultiTerm Extract*

Once the list of candidate terms has been produced by MultiTerm Extract, it is up to the user to select and, if necessary, edit the relevant strings for inclusion in a termbase. A given string might need editing if it contains intervening material. For instance, the string *contribution the public sector makes* should be edited to *make a contribution* before it is stored in a termbase. MultiTerm Extract needs only extract one relevant string for every term (or fixed expression in our experiments), because a single instance of correct identification of a term is enough to ensure inclusion in the termbase.

MultiTerm provides a post-editing tool in the form of scores, as shown in the left column in figure 1 below. Scores consist of a number between 1 and 99 which reflects how confident the system is in proposing that the given string is a term, and they are assigned to every string returned. The higher the number, the more confident the system is. Scores can be particularly useful when dealing with large amounts of output, because they can help speed up the task of seeking out the target items. This only works, of course, if we know under which range of scores we should look for our target items. The difficulty lies in predicting the most useful range of scores in advance, taking into account that scores are not fixed, they vary depending on the parameter settings selected. For terms, we expect scores to be at the higher end, for example 99 is better than 95. For fixed expressions, the empirical question remains as to determining the best range in advance.

In previous research (Fernández-Parra & ten Hacken 2008) we established a trend whereby the majority of fixed expressions appeared to obtain scores ranging from 68 to 75 which, for some combinations of settings, excluded a substantial amount of unwanted strings while only excluding a small percentage of fixed expressions. However, these results were based on a single text and more data are needed to confirm the validity of this trend. Therefore, in this paper we examine the scores returned with additional combinations of settings

with the same source text in order to determine whether the initial trend holds true across other combinations of settings.

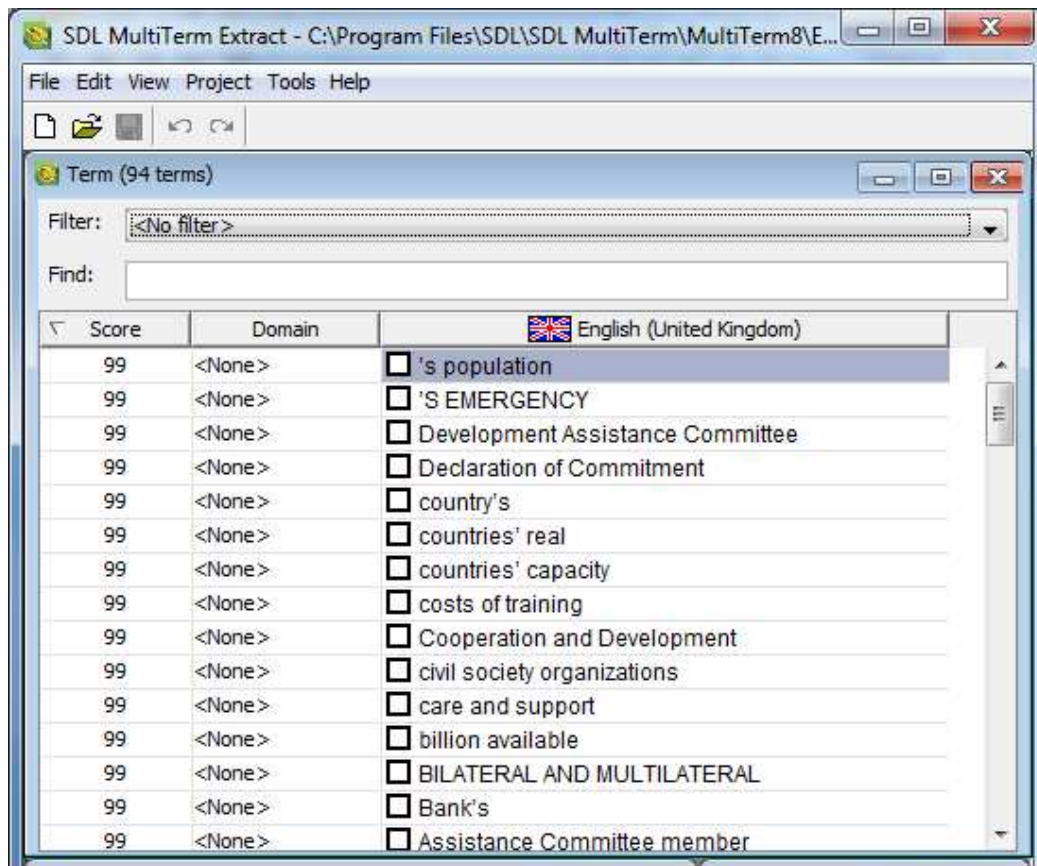


Figure 1: Example of output in MultiTerm Extract

Other post-editing tools in MultiTerm Extract include the creation of a term record and a term validation facility. In the term record, the user can store additional information about the term, such as domain, acronym or definition, before exporting the term to a termbase. Term validation consists of checking the box that appears within each string, as shown in figure 1 above, to indicate the change from candidate term to term. Term records and the term validation facility are not further explored here as these tools do not assist directly in seeking out the target items from the output.

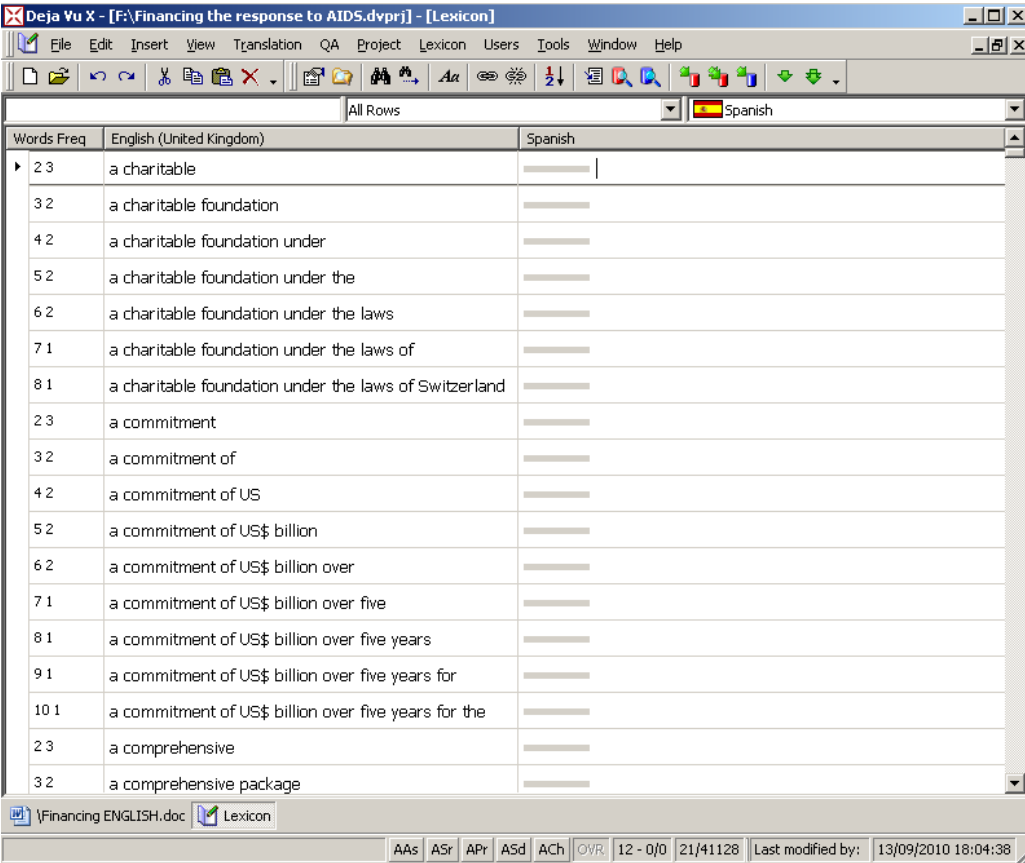
3. Extracting fixed expressions with the Lexicon

The Déjà Vu help files describe the Lexicon as a list of all the words and phrases from the source text. Whereas most CAT tools offer a terminology database and a translation memory, Déjà Vu offers the Lexicon as a third type

of database. The Lexicon has features of term extraction tools and concordancing tools, but is different from both of them in one or more respects.

The Lexicon can be considered a term extraction tool, like MultiTerm Extract, in that it ‘extracts’ and lists potentially useful strings from the source text in order to create glossaries prior to translation.

However, the extraction methods followed by the Lexicon are closer to those of a concordancing tool in that it lists every word and phrase from the source text, without distinguishing between terms and non-terms, or between fixed expressions and irrelevant strings. An example of this is shown in figure 2 below. Although the Lexicon does not automatically display the context a particular string came from, as many concordancers do, the context can be easily viewed by clicking F6.



The screenshot shows the Lexicon software window titled "Deja Vu X - [F:\Financing the response to AIDS.dvprj] - [Lexicon]". The interface includes a menu bar (File, Edit, Insert, View, Translation, QA, Project, Lexicon, Users, Tools, Window, Help) and a toolbar. Below the toolbar, there is a dropdown menu for "All Rows" and a language selector set to "Spanish". The main area is a table with three columns: "Words Freq", "English (United Kingdom)", and "Spanish". The table contains 18 rows of extracted phrases, each with a frequency value in the first column. The Spanish column is currently empty. At the bottom of the window, there is a status bar showing "Financing ENGLISH.doc", "Lexicon", and various system information including "AAs ASr APr ASd ACh OVR 12 - 0/0 21/41128" and "Last modified by: 13/09/2010 18:04:38".

Words Freq	English (United Kingdom)	Spanish
2 3	a charitable	
3 2	a charitable foundation	
4 2	a charitable foundation under	
5 2	a charitable foundation under the	
6 2	a charitable foundation under the laws	
7 1	a charitable foundation under the laws of	
8 1	a charitable foundation under the laws of Switzerland	
2 3	a commitment	
3 2	a commitment of	
4 2	a commitment of US	
5 2	a commitment of US\$ billion	
6 2	a commitment of US\$ billion over	
7 1	a commitment of US\$ billion over five	
8 1	a commitment of US\$ billion over five years	
9 1	a commitment of US\$ billion over five years for	
10 1	a commitment of US\$ billion over five years for the	
2 3	a comprehensive	
3 2	a comprehensive package	

Figure 2: Example of the Lexicon as a concordancer

The Lexicon is not a typical term extraction tool in that it does not provide post-editing features such as term validation, stopwords or scores, or the creation of a

term record. Instead, features such as *Remove entries* and *Sort* can be used, as will be discussed below. In the following sections we describe the typical workflow of the Lexicon and how to adapt it to the extraction of fixed expressions.

3.1. Workflow with the Lexicon

The Lexicon can be created optionally after the source text or texts have been imported into the program, by clicking on *Build Lexicon*, from the main Lexicon menu. The extraction performed by the Lexicon is monolingual, and the Lexicon only has one setting available, *Limit maximum number of words per entry to:*, or entry length, which corresponds to the maximum term length in MultiTerm Extract. Because of the Lexicon's concordance-like methods of extraction, the volume of the output depends only on the maximal entry length and the length of the text. This should be particularly borne in mind when dealing with very large source texts.

Once the Lexicon is generated, the extracted strings can be sorted in a number of ways, such as by frequency, number of words or alphabetically, which will allow the user to collect the items of particular use in one place. Although the Déjà Vu help files only specify the use of the Lexicon as a project-specific or client-specific glossary, the scope of use for the extracted strings is typically threefold. A simplified version of the workflow with the Lexicon is shown in figure 3 below.

First, new terms can be selected from the output, translated and sent to a terminology database or translation memory by using the *Send Lexicon to Terminology Database* and *Send Lexicon to Translation Memory* features respectively. Translated terms can be sent to the relevant database as a batch or individually. In this way, new termbases and translation memories can be created from scratch, or existing ones populated with new items.

In order to establish whether the Lexicon has generated any already existing terms, the *Resolve with Terminology Database* and *Resolve with Translation Memory* features can optionally be used prior to translating any generated strings. This will insert the translations for those strings recognized as already existing in a database. The user can batch delete these items to avoid duplication

of items in the databases. The user can then proceed to translate any new terms and send these to the relevant database as required.

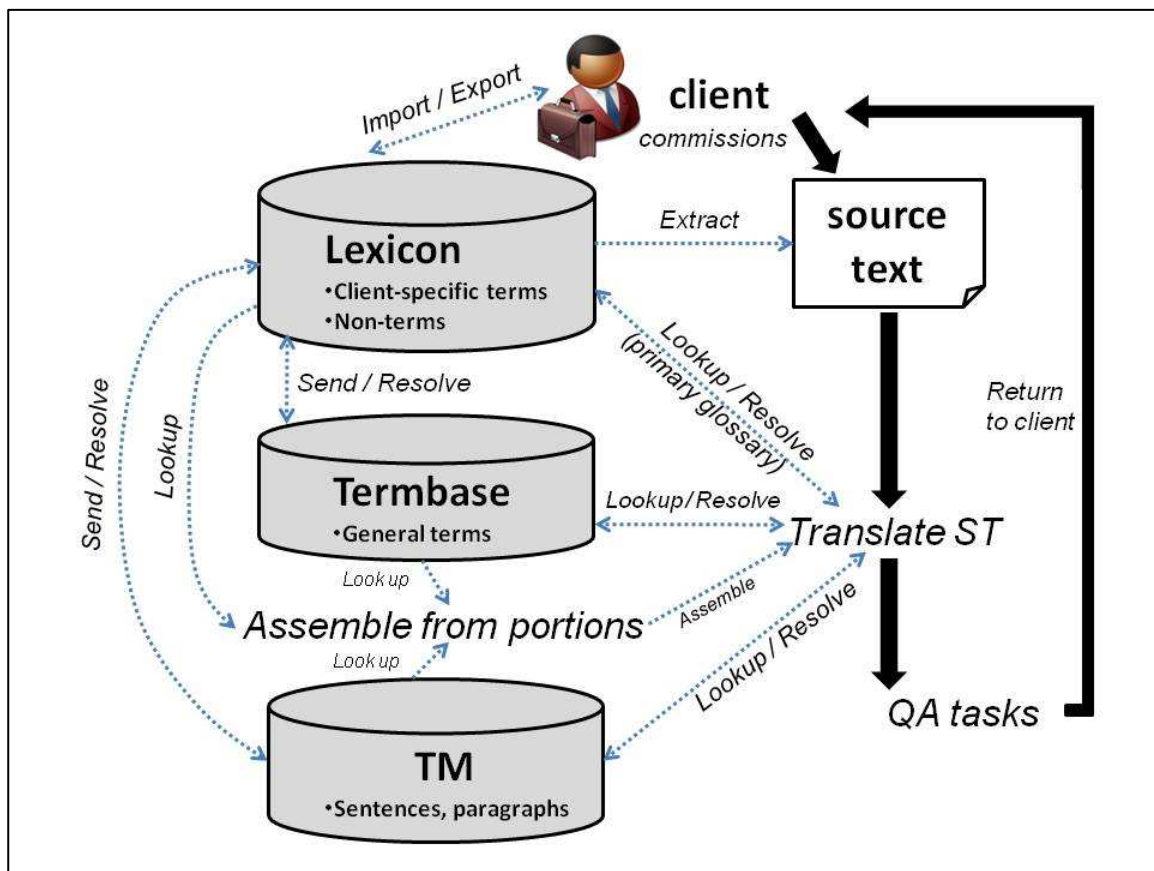


Figure 3: Simplified workflow with the Lexicon in Déjà Vu

Alternatively, the *Resolve* feature can be used as a post-editing tool to translate the contents of the Lexicon. This is a useful tool to create a bilingual glossary, prior to translation, containing only those items from the databases that are specific to the source text to be translated. The Déjà Vu help files suggest that this glossary can become the primary glossary for the project and that it can also be sent to the client, in cases when the client so requests, by means of the *Export* feature. Similarly, if the client provides a project-specific glossary, this glossary can be imported into Déjà Vu as a Lexicon, by means of the *Import* feature.

Secondly, once the new terms have been sent to the termbase or translation memory, they can also be batch deleted from the Lexicon, by means of the *Remove Entries* feature. By now, the items remaining in the Lexicon are typically non-term strings, for example *Check cartridge every*. The Lexicon can

then be used to store all such items. By sorting these items, according to their frequency in the source text for example, the user can make an informed choice as to which of those items might be worth translating, typically those which occur above a certain frequency threshold to be decided on by the user.

The usefulness of a bilingual glossary with these items is to speed up the translation process, as during recognition tasks Déjà Vu will suggest matches from the Lexicon, as well as from the termbase and the translation memory, which can be automatically inserted in the translation. The items that were not translated at this stage need not be deleted from the Lexicon because, during recognition, only those strings with translations will be suggested as matches. Instead, translations may be added to the untranslated strings at later stages in the translation.

Finally, the Lexicon is also used as a reference base when the feature *Assemble from portions* is enabled during translation. Enabling this option means that Déjà Vu will automatically insert relevant words or phrases in the translation by looking up the material not only from the translation memory and the terminology database, but also from the Lexicon. More crucially, with this feature enabled, Déjà Vu will attempt to “turn fuzzy matches into perfect ones by supplying missing terms in whole segments from the translation memory with segments from the terminology database(s) and the lexicon” (Déjà Vu user guide). The more relevant segments contained in the Lexicon, the more successful this automatic translation process will be.

Although Déjà Vu has been in the market for a long time, the different roles played by the Lexicon in different parts of the translation process suggest that the level of sophistication of this tool may be higher than initially thought. Its sophistication is clearly not explicit in the user guide or help files. The conceptual tidiness of keeping non-terms in a separate database from that of terms sets the Lexicon apart from other CAT tools and it allows us to explore the potential of the Lexicon as a tool to deal with fixed expressions.

3.2. *Post-editing tools in the Lexicon*

As well as the *Resolve* feature (cf. section 3.1 above), the *Remove Entries* feature can be used as a post-editing tool. It is accessed by clicking on the

Lexicon tab in the main interface of the program (cf. figure 2 above). The message that appears is shown in figure 4 below.

The first option, *All rows (entire lexicon)*, is useful if the Lexicon needs to be generated again. No further information is given about this option in the user guide.

The second option, *All rows with empty targets*, is useful when the relevant entries have been translated and the rest can then be removed. With the unwanted entries removed, the Lexicon can be sent as a batch to the terminology database or translation memory, or individual portions to each database. This removal can be performed after the first or second stages described in the previous section, if no further use of the Lexicon is envisaged.

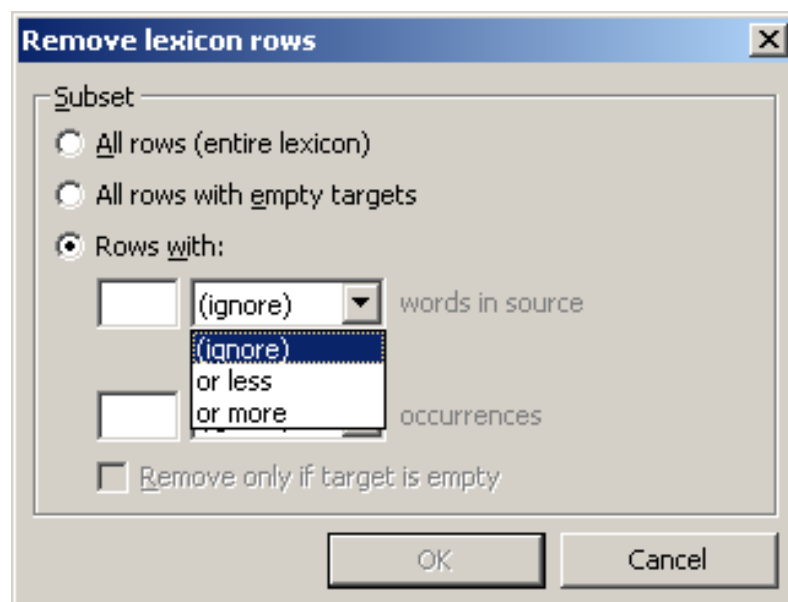


Figure 4: *Remove Entries* feature in the Lexicon

If the Lexicon is to be used in the recognition task during translation, with the *Assemble from portions* feature, as described in section 3.1 above, then it is probably useful not to delete any entries until the translation has been completed, as often more translations are added to the Lexicon strings during translation.

The third option, *Rows with*, constitutes a more selective removal of entries. For example, with the sub-option *words in source*, the user can select to remove all entries which have three words or more, or three words or less. As for the sub-

option *occurrences*, the user can select to remove all entries which occur three times or more in the source text, for example, or three times or less. These two sub-options can be combined with the last sub-option, *Remove only if target is empty*.

Figure 4 above also shows that there is no option to specifically delete translated rows only. Such a feature would be useful in order to delete all the new terms from the Lexicon once they have been translated and sent to the database, thus avoiding duplication. However, there is a way of batch deleting translated rows by marking them as *Finished* (ideally as they are found in the output and edited by the user), and selecting to view finished rows only, as shown in figure 5 below. Then, by simply holding the Shift key, all those rows can be batch selected, and the *Delete* option which appears with a right-click will batch delete these rows. Finally, by selecting to view *All Rows*, the remaining rows in the Lexicon can be accessed again.

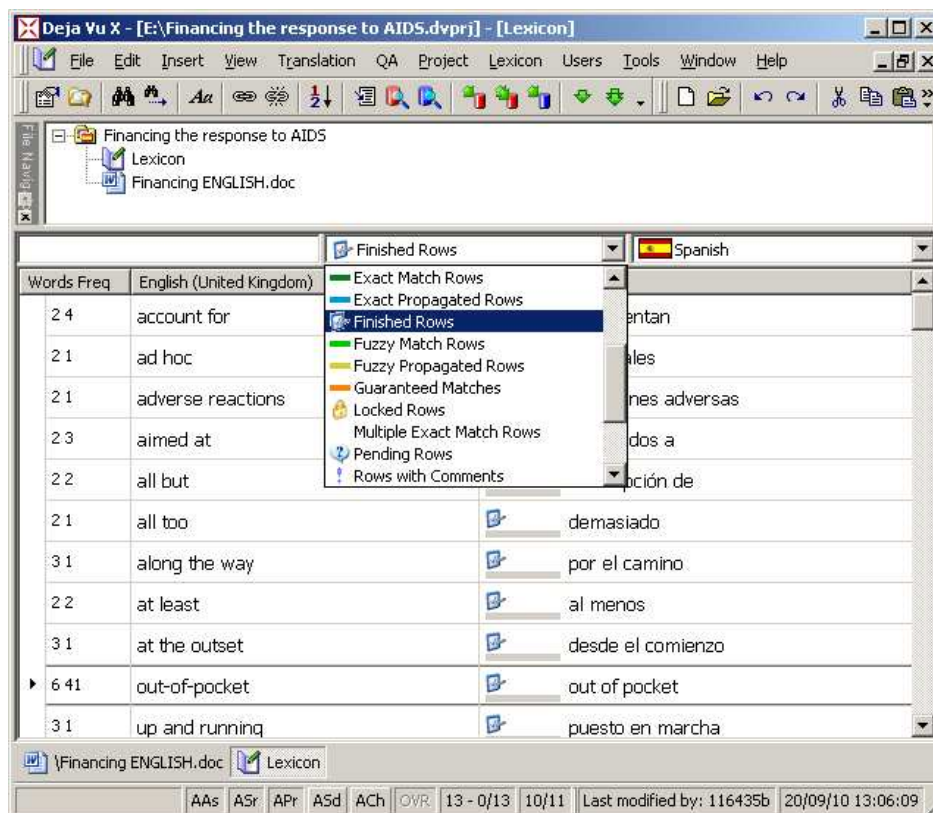


Figure 5: Selecting translated rows for deletion from the Lexicon

4. Setup of the experiments

In our experiments, we focused on comparing the automatic extraction capabilities of the two CAT tools. In total, 54 experiments were carried out, 48 of which with MultiTerm Extract (6 term lengths and 8 noise settings) and 6 with the Lexicon (6 entry lengths, no noise settings available). They were all monolingual extractions because, on the one hand, automatic bilingual extraction is not available in the Lexicon. On the other hand, as reported in Fernández Parra & ten Hacken (2008), bilingual extraction in MultiTerm Extract returns very few valid, usable translated rows for our purposes, and the results for English were not substantially different from those obtained with the corresponding monolingual settings.

The only setting shared by our two CAT tools is maximum term length, (entry length in the Lexicon) or the maximum number of words a string should have. Since we are dealing with expressions consisting of two words or more, we started with maximum term length 2 and we subsequently added maximum term lengths 3, 4, 5 and 6. For comparison purposes, we also added maximum term length 10.

In order to evaluate the results of the experiments, we used standard evaluation metrics in information retrieval such as precision and recall. Because there is an obvious tendency for precision to decrease as recall increases and vice versa, we also used the F-measure, as proposed by Manning and Schütze (1999: 269), which combines both precision and recall to produce a single measure of overall performance.

As a text for all our experiments we used a 10,000-word chapter of the 2006 UNAIDS report, entitled “Financing the response to AIDS”. Prior to all experiments we processed the text manually. During this initial stage we found 90 types of fixed expressions and 220 tokens. The next stage involved determining the individual settings or combinations of settings available in both CAT tools for our task and processing the text once with each setting or combination of settings. Finally, we analyzed the results obtained in the automatic identification tasks against the benchmark of 90 types of fixed expressions.

5. Results of the experiments

The results obtained in our experiments are summarized in table 1 below. The results of each of the experiments with the Lexicon are displayed. For MultiTerm Extract two measures are displayed, corresponding to the highest precision and the highest recall for each of the term lengths, together with the corresponding noise level.

The results shown in table 1 below suggest that for both the Lexicon and MultiTerm Extract, the highest recall with fixed expressions is achieved with longer entry/term lengths. This result confirms the prediction that longer term lengths will obtain better results for fixed expressions because the returned string needs to be long enough for the expression to be ‘embedded’ in it.

ID	CAT tool	Setting	Precision	Recall	F-measure	Candidates
1	Lexicon	Max 2	0.0078	61% (55)	0.0154	7,064
2	Lexicon	Max 3	0.0063	91% (82)	0.0125	13,050
3	Lexicon	Max 4	0.0046	95% (86)	0.0092	18,679
4	Lexicon	Max 5	0.0037	98% (89)	0.0075	23,751
5	Lexicon	Max 6	0.0032	100% (90)	0.0064	28,224
6	Lexicon	Max 10	0.0022	100% (90)	0.0044	41,128
7	MTE	Max 2, noise 0.2	0.0179	1% (1)	0.0137	56
8	MTE	Max 2, High	0.0146	24% (22)	0.0276	1,505
9	MTE	Max 3, noise 0.1	0.0714	3% (3)	0.0455	42
10	MTE	Max 3, High	0.0242	62% (56)	0.0466	2,313
11	MTE	Max 4, noise 0.1	0.0789	3% (3)	0.0469	38
12	MTE	Max 4, High	0.0253	71% (64)	0.0489	2,530
13	MTE	Max 5, noise 0.1	0.0811	3% (3)	0.0472	37
14	MTE	Max 5, High	0.0296	80% (72)	0.0571	2,434
15	MTE	Max 6, noise 0.1	0.0811	3% (3)	0.0472	37
16	MTE	Max 6, High	0.0327	82% (74)	0.0629	2,263
17	MTE	Max 10, noise 0.1	0.0811	3% (3)	0.0472	37
18	MTE	Max 10, High	0.0462	85% (77)	0.0876	1,668

Table 1: Summary of results obtained (MTE = MultiTerm Extract)

With the Lexicon, full recall is achieved, but it should be noted that, in order to find our 90 fixed expressions, we had to search through over 40,000 rows of candidates at worst. The first entry length to obtain full recall is Max 6 with the Lexicon. The question remains as to how to determine in advance the optimum entry length for a new text. We were able to retrieve our targets relatively

quickly because we knew in advance which items we were looking for. Therefore, sorting the output alphabetically facilitated our retrieval task. However, in the case of using the Lexicon as a term extraction tool to find new items, we would not have a list of such items in advance, so the post-editing task would be considerably slower.

With MultiTerm Extract, the highest recall is somewhat lower (85%) than with the Lexicon, but this is still a reasonable result, especially if we consider that the volume of output in which to find our 90 fixed expressions is some 40 times smaller than that of the Lexicon. This is probably why the highest F-measure (0.0876) is achieved with MultiTerm Extract.

The highest precision (0.0811) is also achieved with longer term lengths in MultiTerm Extract but with a very low level of noise (0.1), so that recall is extremely low (3%). Precision is extremely low in all of the experiments with the Lexicon. If we look at sacrificing some precision for the sake of increasing recall, by searching through the highest F-measures achieved in our experiments, we find that the High noise level is the setting with which the most reasonable results are returned, e.g. in projects 12, 14 and 16 in table 1 above.

In the following sections we give an overview of the results obtained with the Lexicon and MultiTerm Extract. We also discuss the main problems we encountered when applying software that was specifically designed with terminology in mind to the identification of fixed expressions, the role of frequency in the source text, the variation of the expressions and suggest ways of retrieving fixed expressions from the output with both CAT tools.

5.1. Results obtained with MultiTerm Extract

With MultiTerm Extract, as with the Lexicon, the best recall is obtained with higher term lengths, for the same reason that many of our fixed expressions appear embedded in longer strings within the source text. Therefore, in order to identify the expression, the term length setting has to be long enough to cover the string where the expression appears.

Because in all of our experiments we applied the stopword list and the basic vocabulary list, full recall is not achieved in the projects shown in table 1 above. Without such lists, recall with MultiTerm Extract can be 100% if term length 6

or higher is selected, at least with our particular source text. However, we found that, without stopword and basic vocabulary lists, the volume of the output is much larger, from four to seven times larger, making post-editing tasks considerably slower. We also found that scores did not especially facilitate the post-editing tasks in this case, because most of the strings had been assigned scores within the range where our target items were found, in the region of 62 to 78. Taking this into account, applying stopword and basic vocabulary lists can be considered useful when identifying fixed expressions because recall remains relatively high and post-editing tasks can be performed more effectively than with a larger output.

5.2. Results obtained with the Lexicon

With the Lexicon, it is possible to identify all of our fixed expressions. However, in order to achieve full recall, the entry length setting has to be set at 6 or higher, given the embedded occurrences of the fixed expressions within longer strings in our particular source text. We cannot expect to achieve full recall with the Lexicon if the entry length is set at 2 because, as we have seen in section 2.1 above, this setting would exclude expressions such as *honour a commitment*. This is why entry lengths 5 and lower do not achieve full recall. Once full recall has been achieved with a particular length n , it will also be achieved for any value larger than n . This is easy to see because for each value n , the Lexicon properly includes the entire Lexicon generated for $n - 1$. As noted above, the crucial question remains to find the minimal value for n such that all fixed expressions are found.

In the Lexicon experiments, we were interested in obtaining the best results with the lower entry lengths because, as we have seen in section 3.1 above, the volume of the output increases considerably as entry length increases. Entry length 6 produced over 28,000 strings for our 10,000-word source text. Therefore, it is worth investigating further the results obtained with entry lengths 3, 4 and 5, because the volume of the output was significantly lower than with entry length 6, while recall remained relatively high. The question raised here is how our target items can be best retrieved from a large output with the current features of the program.

5.3. *Frequency of terms vs. frequency of fixed expressions*

Whereas terms can be expected to be significantly more frequent in specialized texts of their relevant domain than in non-specialized texts, there is no reason to expect significant variation for fixed expressions. The role played by frequency is much more salient in the MultiTerm Extract approach than in the Lexicon approach, because MultiTerm Extract analyzes the frequency of the words within the source text and extracts those items which have a higher frequency than a given threshold. The Lexicon follows a ‘bag of words’ approach, where every word and phrase is extracted regardless of its frequency in the source text, though the frequency is listed next to every extracted string.

Therefore, with the Lexicon we can expect all the expressions to be identified. In our experiments, this was the case with entry lengths 6 and higher. With MultiTerm Extract, we can expect higher identification rates for those expressions which occur a certain number of times in the source text. In our experiments, this was borne out by the fact that the only five expressions which occurred more than five times in our source text were identified with most of the combinations of settings and were often assigned scores of 99. For example, the expression *response to* occurred 23 times in our text and scored 99 with every combination of settings in our experiments.

Those expressions which only occurred once in the source text were identified with fewer combinations of settings and the scores they obtained in MultiTerm Extract typically ranged between 59 and 73. For example, the expression *raise money* occurred only when High noise settings were applied and its scores ranged from 65 to 76. Furthermore, it is interesting to note that generally the lower scores corresponded to the shorter term lengths and the higher scores to the longer term lengths.

5.4. *Degree and type of variation of fixed expressions*

Automatic term extraction tools are often pattern-based extraction systems which look for certain recurrent word combinations or associations within a source text (cf. Heid 2006). This is because terms are often stable nominal groups and have restricted freedom of order and insertion. Fixed expressions, however, have considerably larger freedom in this respect. Therefore, it is not surprising to find an expression such as *meet a requirement* in the string *If the*

funding requirements for 2006–2008 (shown in Figure 10.1) can be met, where the word order of the component words of the expression is not only reversed, but there is also intervening material between them. By contrast, we would not normally expect to find intervening material between the component words of a term such as *ink splitting mechanism*.

The variation of fixed expressions poses a challenge to the identification of such expressions, especially by statistical term extraction methods, as with MultiTerm Extract. In concordance-like approaches to extraction, such as in the Lexicon, the variation of fixed expressions does not constitute an obstacle to their identification, because neither linguistic knowledge nor statistical probabilities are applied.

Of the 90 expressions in our text, 42 expressions (47%) had some type of variation. They differed from the base form by number and tense variations, intervening material, inversion of the word order, or a combination of these. The type of variation that is most likely to affect recall in MultiTerm Extract as a statistical tool is intervening material, because the other types of variation presuppose access to linguistic rather than statistical knowledge. The degree of intervening material in our experiments can be measured in terms of how many words are found between the component words of the expression. Thus, out of the 42 fixed expressions with variation, 18 (42%) had intervening material, which ranged from zero to nine words. For example, the component words of the expression *make use of* are separated by two words in the string *making far better use of funding*.

Our initial findings suggest that the higher degree of intervening material in a given expression, a longer term length setting is needed for the correct identification of the expression. This means that, although term length 2 should be theoretically sufficient to identify a 2-word expression such as *in need*, if it occurs in a string such as *80% of those in urgent need*, term length 3 would be needed in practice to identify this expression. By contrast, in the case of terms, such as *ink splitting mechanism*, we would expect that term length 3 would be enough both in theory and in practice to identify the term.

Our results suggest that the variation of fixed expressions, particularly in the form of intervening material, is one of the main causes of non-identification of

the expressions. However, it appears that the expressions found in our source text did not in fact vary as much as expected, as only 14 expressions (15%) had two or more words of intervening material. The question remains as to how to measure the degree of intervening material within the fixed expressions of a given text prior to extraction with MultiTerm Extract, so that we can establish in advance which term length will produce optimum results.

5.5. Retrieving fixed expressions with MultiTerm Extract

In this section, we evaluate the usefulness of post-editing tools in MultiTerm Extract to extract the target expressions from the returned output, in particular by using scores (cf. also section 2.2 above). The results from our previous research (cf. Fernández-Parra & ten Hacken 2008) with MultiTerm Extract pointed towards an optimum range of scores being from 68 to 75, where the bulk of fixed expressions would be found. It was an optimum range because it excluded 26% of the noise but only 5% of the expressions. Our research was made up of experiments with 192 different combinations of settings (term lengths and noise settings). In the current research, we added a further 54 combinations of settings that we had not tested before with the same source text.

Scores can be grouped, presented and interpreted in a number of ways, as there are so many variables involved. Therefore, we focused on the analysis of the 111 scores assigned to results from experiments with term length 10 because it has the largest concentration of fixed expressions. From the eight extractions we carried out with term length 10, we focused on the project which combined this term length with High noise level, because this project yielded the highest recall in MultiTerm Extract. The distribution of scores assigned to fixed expressions in this project is shown in table 2 below. This project is project 18 in table 1 above which returned 1,668 candidate terms and identified 77 expressions.

The results shown in table 2 below suggest that, if we take project 18 as a model, the optimum range of scores in our experiments is from 72 to 78. If we only looked within this range of scores, we would find 68 fixed expressions (75%) and we would only need to search through 1,114 candidates instead of 1,668. This constitutes a 33% reduction (554 fewer candidates) of the output to search through.

This range of scores is somewhat narrower than that of our previous findings, 68 to 75, but it corresponds to only one project. If we extend the analysis to all of our current experiments, the range widens to 62 to 78, maintaining similar statistics in the reduction of the output at the expense of some fixed expressions.

No. of fix.expr.	Score	No. of cand.
0	47	146
2	50	100
0	54	2
2	59	108
1	65	74
0	69	79
3	72	100
6	74	126
12	76	280
50	78	608
0	85	2
0	90	3
0	91	4
0	93	1
0	94	2
1	99	33
77	-	1,668

Table 2: Distribution of scores in project 18 (Max 10) in MultiTerm Extract

5.6. *Retrieving fixed expressions with the Lexicon*

With the Lexicon, we focused on project 6 (cf. table 1 above), in which we applied entry length 10, because this project produced the highest number of candidates (41,128) as well as recall (100%). In our experiments it was relatively simple and fast to retrieve the 90 fixed expressions from such a large output by sorting the Lexicon alphabetically, because we knew in advance which expressions we were looking for. However, we also attempted the post-editing task without making use of that a posteriori knowledge, so that we could evaluate to what extent the Lexicon can be used as an extraction tool for fixed expressions. In this section, we report on the results of these attempts.

We found two different approaches to the retrieval of the expressions which involved the use of the *Remove entries* and *Sort* features, and which reduced the

output by similar amounts. The first approach we followed was to use the *Sort* feature to sort the output alphabetically, without making use of the *Remove entries* feature. We found that with this approach, the contents of the output were displayed as in a concordancer, as displayed in figure 6 below. Segments with similar text are sorted vertically in groups of ten. This is shown in figure 6 below from the segment *up* to the segment *up and running required setting up management structures*, from which the segment *up and running* is selected as containing a fixed expression. The reason segments are grouped in ten at a time is because the entry length for this project was set at 10.

The screenshot shows a window titled 'Deja Vu X - [E:\Financing the response to AIDS.dvprj] - [Lexicon]'. The window contains a concordancer interface with a menu bar (File, Edit, Insert, View, Translation, QA, Project, Lexicon, Users, Tools, Window, Help) and a toolbar. Below the toolbar is a table with the following data:

Words Freq	English (United Kingdom)	Spanish
6 4	until the following year or that	
7 1	until the following year or that may	
8 1	until the following year or that may be	
9 1	until the following year or that may be spent	
10 1	until the following year or that may be spent over	
1 11	up	
2 1	up and	
3 1	up and running	
4 1	up and running required	
5 1	up and running required setting	
6 4	up and running required setting up	
7 1	up and running required setting up management	
10 1	up and running required setting up management structures and employing	
9 1	up and running required setting up management structures and	
8 1	up and running required setting up management structures	
2 1	up as	
3 1	up as a	
4 1	up as a charitable	
5 1	up as a charitable foundation	
6 4	up as a charitable foundation under	

Figure 6: Example of retrieval from a concordance-like output¹

¹ It is not clear why the string *up and running required setting* is listed with a frequency of 1 only, whereas the string *up and running required setting up* is listed with a frequency of 4, and why the strings with length 10 and 9 are not after the one with length 8.

This approach does not necessarily reduce the size of the output but, because we are essentially looking at ten segments at a time, it would theoretically take ten times less to process the whole output, as if the output only consisted of 4,112 items instead of 41,128. The strength of this approach lies in the visual ease of lookup of such a display and the fact that many expressions (48% in our experiments), as in the case of *up and running* in figure 6 above, will not need further post-editing.

The second approach consisted of using the *Remove entries* feature to leave only 6-word strings in the output, as we found that entry lengths 5 or lower did not achieve full recall. To this end, the *Remove entries* feature needs to be run twice, for example by first removing rows with 5 or less words in source, and then by removing rows with 7 or more words in source. The reason for this is that the option of removing strings by number of words can only be used once at a time, as shown in figure 4 above. This approach reduces the output to 4,429 rows, roughly ten times smaller than the initial 41,128 rows. An example of the output reduced in this way is shown in figure 7 below.

Words Freq	English (United Kingdom)	Spanish
6 4	United States support a level of	
6 4	universal access to treatment by for	
6 4	universal access" means in different countries	
6 4	universal access" occurs when % of all	
6 4	until the epidemic is stopped and	
6 4	until the following year or that	
6 4	up and running required setting up	
6 4	up as a charitable foundation under	
6 4	up comprehensive prevention programmes and the	
6 4	up efforts to monitor and evaluate	
6 4	up management structures and employing coordinators	
6 4	up national responses and how the	
6 4	up of the AIDS response in	
6 4	up to % of HIV transmission and	

Figure 7: Example of retrieval from a non-concordance-like output

With this approach, the output can be sorted alphabetically as with the first approach but, because we removed all similar strings by removing shorter and longer strings than our selected entry length, the output does not have the same degree of visual ease as the first approach. This is shown in figure 7 above, where the string containing the same expression *up and running* has been selected for retrieval. Although with this approach the volume of the output has been reduced, post-editing tasks might take longer because every expression retrieved in this way will need editing before exporting to a termbase. Longer entry lengths imply larger volume of intervening material in every string. For example, in figure 7 above, the string *up and running required setting up* will need to be edited to *up and running*.

Although each approach has advantages and disadvantages, our results suggest that, in practice, we only needed to post-edit a tenth of the output returned with the Lexicon and still achieve full recall. Therefore, precision in the Lexicon is maximized during post-editing, but it should be stressed that the optimal entry length 6, the lowest to achieve full recall, is not given in advance.

6. Conclusions

In this paper, we compared the automatic extraction capabilities of MultiTerm Extract and the Lexicon when applied to the extraction of fixed expressions. It should be borne in mind that neither CAT tool was designed for such purpose, therefore any limitations presented here, or our conclusions in this respect, should not be taken as an assessment of the software. Similarly, we should point out that our experiments were exploratory in that we focused on a single text so that the potential of each CAT tool for our special purpose could be looked at in as much detail as possible. Ideally, the trends found in our experiments should be further tested on a larger amount of data.

The automatic extraction approach followed by each CAT tool is different. Whereas MultiTerm Extract is a statistical term extraction tool, the Lexicon works rather like a concordancer. In other words, in our experiments we compared the selective recall approach of MultiTerm Extract to the ‘bag of words’ approach of the Lexicon applied to the extraction of fixed expressions. A number of conclusions can be drawn from our experiments.

- Full recall is achieved with the Lexicon, but with extremely low precision. We found that we could eliminate noise efficiently during the post-editing stage by means of two different post-editing approaches that reduced by approximately 90% the volume of the output. With MultiTerm Extract, full recall could be achieved if exclusion files were not applied. However, we found that it was useful to sacrifice some recall by applying both default exclusion files for the sake of quicker post-editing tasks.
- With both CAT tools, the best results were obtained by selecting longer term lengths (Max 6 with the Lexicon, Max 10 with MultiTerm Extract), because of the variation of fixed expressions, whose component words do not always occur adjacently in the string, but often ‘embedded’ in it.
- Scores were a useful post-editing tool in MultiTerm Extract because they narrowed the scope of search for expressions within the output. The optimum range of scores across all our experiments is 62 to 78. However, it is difficult to predict the optimum range in advance. With the Lexicon, we used the *Resolve*, *Sort* and *Remove entries* features as post-editing tools. The *Resolve* feature allowed us to leverage the contents of the Lexicon against existing databases. The *Sort* feature was useful in that it considerably reduced the time devoted to post-editing because many fixed expressions would not need further post-editing, although with this feature the volume of the output was in fact not reduced. The *Remove entries* feature reduced the volume of output but all fixed expressions retrieved in this way would need further post-editing.
- The three main factors that affect the recall of fixed expressions with MultiTerm Extract are the frequency of the expressions in the source text, the presence of the expressions in exclusion files and the variation of the expressions. These factors had no bearing on the results obtained with the Lexicon.
- The difficulties encountered by MultiTerm Extract and the Lexicon in the identification of fixed expressions seem to stem from the differences between such expressions and terms. Term extraction software counts on the stability of the term as a unit and on the frequency of that unit in the

text as anchors for the identification of the terms. Fixed expressions are more variable and typically less frequent in the text than terms.

Although the highest F-measure (overall performance) corresponded to MultiTerm Extract, the Lexicon proved to be a much more sophisticated tool than realized by many. The relative merits of each approach warrant further research on a much wider selection of source texts.

7. References

- Bowker, Lynne (2002) *Computer-Aided Translation Technology: A Practical Introduction*. Ottawa: University of Ottawa Press.
- Colson, Jean-Pierre (2004) “Phraseology and Computational Corpus Linguistics: From Theory to a Practical Example”. In H. Bouillon (ed.) *Langues à Niveaux Multiples*, Louvain: Peeters, pp. 35-45.
- Coulmas, Florian (1979) “On the Sociolinguistic Relevance of Routine Formulae”. *Journal of Pragmatics* 3: 239-266.
- Fernández-Parra, María & ten Hacken, Pius (2008) “Beyond Terms: Multi-Word Units in MultiTerm Extract”. Proceedings of *Translating and the Computer 30*, London: ASLIB.
- Heid, Uli (2006) “Extracting Term Candidates from Recursively Chunked Text”. In P. ten Hacken (ed) *Terminology, Computing and Translation*. Tübingen: Gunter Narr, pp. 97-116.
- Manning, Christopher & Schütze, Hinrich (1999) *Foundations of Statistical Natural Language Processing*, Cambridge, MA: MIT Press.
- Mel’čuk, Igor (2001) “Collocations and Lexical Functions”. In A.P. Cowie (ed.) *Phraseology. Theory, Analysis and Applications*. Oxford: Oxford University Press.
- Pulman, Stephen (1993) “The Recognition and Interpretation of Idioms”. In C. Cacciari and P. Tabossi (eds.) *Idioms – Processing, Structure and Interpretation*. NJ: Lawrence Erlbaum, pp. 249-270.
- Sag, Ivan; Baldwin, Timothy; Bond, Francis; Copestake, Ann; Flickinger, Dan (2002) “Multiword Expressions: a Pain in the Neck for NLP”. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2002)*, pp. 1-15.