JEAS | Journal of Economics and Administrative Sciences
E-ISSN: 2148-1792 P- ISSN: 1302-2024
Volume 5, Supplement Issue 1
www.jeasweb.org

Translation Memory: Towards Software Localization

Ladan Azadeh
Foreign Languages Department, Payame Noor University, I.R of Iran

Tayebeh Mosavi Miangah
Linguistics Department, Payame Noor University, I. R. of Iran.

ABSTRACT

Translation Memory (TM) is a way of collecting, storing and reusing translations. It is a computer tool with the aim of providing facilities to reuse the existing translations. In its simplest form it contains a database in which translator can store previous translations for future use and easy search. As the raw output of automatic Machine Translation (MT) systems are not very reliable in most areas, in recent years there seems to be an increasing tendency among the translators towards using computer assisted tools like TMs.

In this study we tried to localize a TM system called MetaTexis to be applicable for translation from English into Persian. To examine the potential of this TM system, the required adjustments on the system were made and the performance of the system was evaluated through an experiment using five different text types, including literary, law, political, religious and technical texts. The results of the experiment show that this TM system works well on technical texts, while the highest rate of translation errors belongs to the literary texts. The total accuracy of the system performance for the English and Persian language pair achieved 90.78 percent which is encouraging for the pair.

Keywords: Bilingual parallel corpus, Computer-assisted translation, English-Persian translation, Software localization, Translation memory.

1.Introduction

The increasing expansion of communication means and the emergence of Internet along with the disability of traditional approaches towards translation as well as insufficiency of machine translation methods on one hand, and raising state and private demands for translation and the shortage of time due to the competitive ground in the international markets on the other hand lead to the formation of the idea of using the previously done translations for providing new translations with the help of information retrieval and information storage as well as corpus linguistics.

The availability of huge volumes of bilingual digital texts in different forms along with various aligning devices made it possible to construct a reliable database for the translation memory systems. Apart from the others, this is one reason for changing tendencies from automatic translations towards translation tools - systems which support the translators instead of taking the place of them

Translation Memory (TM) can be considered a special parallel corpus in which the aligned texts are stored in a database and then a lexical unit along with its previous translational equivalents can be retrieved in case it is required to be re-translated (Mosavi Miangah, 2008). TM is a way of collecting, storing and reusing translations. TM systems are sets of computer tools with the aim of providing facilities to reuse the existing translations. In TM the purpose is systematic archiving the output product of the translators in pairs of matched chunks from source and target languages (Kumar, 2005).

In contrast to machine translation, a TM does not aim at producing automatic and correct translation, but it tries to help translator in automating some parts of a translation task. Solving lexical problems in translating texts in huge volumes is the main purpose of TM systems. Repeating a sentence or part of it in a long translation text is not only time-consuming but very tiresome wasting time and energy. It is even possible that a part of sentence or phrase is skipped. TM removes the probability of facing such event and this is a great advantage in its own right.

The TM software which is increasingly used by companies and individuals involving translation in order to accelerate the translation process while decreasing its expenses entered into the market in early 1990. TM can lessen editing and processing expenses to a high degree, prevent from translating repetitive parts of the documents under translation, and decrease the rate of redundancy. TM which now makes possible to create adjust, store, and access the existing translations for the subsequent uses, is said to be the most prominent assisting tool for professional translators.

IBM, one of the pioneers of translation memory software, used to spend more than $100 million a year to translate its documentation and localize its software. To reduce its costs, it developed a translation memory program to handle its own work. The program, called Translation Manager 2, is now sold to other companies (Freivalds, John. 1999). TRADOS is a German company that is well known for its translation tools including translation memory ones. Section 8 introduces some translation memory systems available in the market now. The end-user of translation memories are translators, companies with in-house translation divisions, translation agencies and direct clients.

Translation memory is not a new concept. The academic research on this area began in the late of 1970s, and the first commercial products were released into the market just in the late 1980s. However, translation memories found their commercial justification only from the late 1990s.

The original idea is usually attributed to Martin Kay and his 1980 paper though the TM idea was not clearly addressed. "... the translator might start by issuing a command causing the system to display anything in the store that might be relevant to… Before going on, he can examine past and future fragments of text that contain similar material" (Kay, 1997: 19).

Arthern explain what we now call a TM system more explicitly: "It must in fact be possible to produce a program which would enable the word processor to 'remember' whether any part of a new text typed into it had already been translated, and to fetch this part, together with the translation which had already been translated … Any new text would be typed into a word processing station, and as it was being typed, the system would check this text against the earlier texts stored in its memory … One advantage over machine translation proper would be that all the passages so retrieved would be grammatically correct. In effect, we should be operating an electronic 'cut and stick' process which would, according to my calculations, save at least 15 per cent of the time which translators now employ in effectively producing translations." (Arthern, 1981: 318). The idea was also used in Alps one of the first commercial systems developed in Birmingham Young University which was called 'Repetitions Processing' and was able to retrieve only exact matches. As Melby says, the source code of this system was then used by IBM in its well-known system, Translation Manager (Melby, 1995). In IBM-style alignments, IBM models 1 to 5, a single target word can be connected to several source words. Alignment models proposed by Melamed and Wu allow "one-to-one" alignments (Melamed, 1998, and Wu, 1997).

Planas proposes an approach for a Sub-Sentential Translation Memory which is based on sequences of syntactic chunks, as defined by Abney (1991). The contents of the TM and the new text are segmented into chunks; sequences of chunks from the new text are searched in the TM and the translation of the matched sequences will be proposed to the user as partial translations of the current input (Planas, 2000). Similarly, Simard and his colleague Langlaise tried to evaluate the potential of a type of translation memory system

capable of supplying a human translator with sub-sentential segments of target language text. They proposed an architecture based on a more flexible searching mechanism than found in existing TMS's. Their experiments indicated that the new strategy could produce substantial improvements in recall, while maintaining precision at reasonable levels, especially when the text to be translated was related to the content of the translation memory. But in the end, they concluded that their research showed that existing TMS's were extremely far from exploiting the full potential of translation memories and that finding better ways of extracting text at the sub-sentential level turned out to be a promising avenue (Simard and Langlaise, 2001).

In another study, Yamada tried to investigate how productivity is affected by different kinds of TM databases. The impact of two different versions of a TM database – free vs. literal TMs - was examined through a pilot study. The results showed that in the higher fuzzy-match categories, translators using the less literal TM did not gain as much speed as was the case when using a more literal TM (Yamada, M, 2011).

There are many translation memory systems, some of which are free and some others need to be subscribed to be used. The free TM systems already available are Omega t+, GPL-application written in Java, Transolution, GPL-application written in Python, and Open Language Tools, a TM application supporting XLIFF files, open sourced by Sun. some other TM systems are Trans Suite 2000 (Cypresoft), IBM TransLexis, MetaTexis, MultiTrans of MultiCorpora, Passolo, and Translator's Workbench.

In this study the main attempt was to localize a TM system called MetaTexis to be applicable for translation from English into Persian. To examine the potential of this TM system, the required adjustments on the system was to be made in order to be suitable and applicable for English and Persian language pair. The idea of localization has been around since the early 1980s when the first localization projects were undertaken. As the sales and marketing experts of large US-based IT developers in the 1980s looked for opportunities to grow sales outside of their native US-market, they saw Europe as their next major market. From this moment the localization industry was born. While people living in large non-English European economies was able to purchase the expensive computing hardware and software but they could not use them in English, they felt a need for those word processors, spreadsheets and, even presentation software to be translated and adapted for meeting their needs (Yuste, Rodrigo, 2008). The experiment of this study has been carried out using the trial version of MetaTexis, version 4.0, 2011, with many new functions. MetaTexis runs under Microsoft Word and comprises all functions of a professional CAT tool like TRADOS or DejaVu. It is comparable to Wordfast. However, MetaTexis follows a different technological approach and puts special emphasis on ease of use and detailed statistical information for translators.

2. Classifying Translation Memory Systems

Most TM tools are without database which is the main element of each TM system. Creating such database is conceded to the users, while the systems themselves act as a framework for preserving and using such a memory. This way, the system stores and indexes the previously translated contents in an organized way so that in subsequent stages when searching by the user, it can retrieve maximum amount of information out of these contents (Lagoudaki, 2006). The main difference between TM systems is in the ways used for data processing. Such data processing contains segmenting, aligning, indexing and matching. In most TM systems the processes related to segmenting and aligning are followed by text indexing.

A TM system divides the source and target texts into segments. These segments form translation units which may be a whole paragraph, sentence or phrase. Then, these units are aligned and stored in the system's databank as indexed translation units. Some TM systems store the texts as complete bi-texts and index them using character-string-in-bitext (CSB) technique (Gow, 2003) and then align them at paragraph level in the databank. The two techniques for retrieving counterparts in recent commercial TM systems are match-finding based on character sequence and match-finding based on language (Lagoudaki, 2006). TM tools which use the character-based approach try to search the counterparts not only at the segment level but at the parts of a segment existing in the database.

In order to use the second approach some linguistic information should be introduced to the system. In such systems after segmenting both source and target texts, they are linguistically analyzed at sentence level

and the sentences are divided into syntactic chunks and some grammatical annotations are added too. This is usually done with the help of monolingual dictionaries and the algorithms capable of determining grammatical classes. As in this approach the grammatical category of the chunks are determined, the system search for those chunks with similar grammatical categories, and this way, the precision of the system is to be increased.

Each of two above-mentioned approaches has its own advantages and disadvantages. As the character-based approach is a language independent one, it can be well applied to almost unlimited pairs of languages. However, the advantage of the language-based approach is its improved recall and segmenting precision (Lagoudaki, 2006).

Another distinction between TM systems is in the environment they present to the users to translate. Some TM tools act as an add-in or a macro in Microsoft Word environment and enable the user to process and translate in this word processing environment, while many other systems have their own specialized environment for editing or processing. Some examples of the first group are 'WordFast', 'MultiTrans', 'Logoport', Metatexis', 'Trados', and 'Fusion'. 'Dejavoo', Heartsome', 'MemoQ', 'StarTransit', 'SDL' and 'Across' are those belonging to the second group. Each of these tools has its own properties and advantages, and the choice is up to the user.

Another classification we can make on TM systems is that some systems are open source and some others non-open source. The cases like 'Trans solution', 'faren desk', 'OmegaT' and 'Open Language Tools' are among open source TM systems the two last ones are frequently improving.

The idea of using the previously translated units and the hidden information within them has a rather long story and it was firstly found in the forms of example-based and statistical-based machine translation systems. Statistical systems indirectly make use of previously translated units. These systems extract the information available in translated units for patterning the system using statistical analyses and probabilities. Example-based systems, however, make more direct use of such units. As accessing to automatic example-based machine translation like other systems did not lead to an absolute success, the idea of using translation tools and designing interactive environments was introduced (Ahrenberg and Merkel, 1996).

Translation memories are, in fact, related to example-based and statistics-based machine translations from the working idea standpoint. Translation memories generally work on documents whose similar contents were entered into the memory of the system. Statistics and example-based translations tend to make it easier for translators to translate unseen texts by using the previous translations. This way, they are approaching to a form of experience-based machine translations.

Translation memory in its simplest form contains a database in which translator can store previous translations for future use and easy search. Machine translation translates linguistic data from one language into another based on grammatical rules and dictionaries, while translation memory does not do translation activity in its own right but concedes it to the translator. Translation memories, in fact, inscribe pairs of linguistic units unto the memory and search the previous translation units and suggest them upon meeting similar units while presenting a text for translation. The suggestion may be accepted, edited or rejected. Most programs use fuzzy matching algorithm in this respect. The revenue of using such programs may not be conceivable at first point. It may seem first that the software steps down the process of translation instead of accelerating it, but the advantages will be revealed in the course of time. The memory will be enriched by accumulating translated texts and the found matches will be more precise too.

There are many similarities between TM packages. The search is often carried out not only for complete chunks or sentences but for words and phrases. This may help the translator find the meanings of words or phrase in order to be more consistent in terms of lexical units. Almost all TM programs have a section for terminology management. They can transfer the files between other systems of their type. Some TM programs work as gusts within a word processor like Microsoft, and most of them have their own special editors. All such packages have some filters to exchange files with different formats, and some have an alignment tool for adding past translations to the memory for reusing.

Translation Memory Applications

Translation memories are often used in combination with tools of computer-assisted translation, word processing programs, terminology management systems, multi-lingual dictionaries or even machine translation systems and their raw outputs. Most companies which produce multilingual documentations make use of TMs; however, TMs can be found suitable devices for independent translators too.

Today the manuals, brochures, web sites, email messages, and internet chats constitute a huge part of translation market in which TM can be well applied due to the type of content. Among the text types in which TM show the best function are the following cases:

a) Large technical texts like manuals, instructions for application and maintenance of tools and machinery containing repetitive lexical units

b) Comparable textual material like user interface in different software, help files in software and websites

c) The documents and texts which are frequently updated due to their special characteristics like weather reports

Generally speaking, in the following cases TM systems are not only a suitable choice, but also necessary tools for translators: the number of repetitious chunks in texts under translation is high, the consistency in equivalent-finding is emphasized, it is cost-effective, there is no alternative case to be substituted, there are suitable data (the previously translated texts very similar to the texts under translation in appropriate formats (electronic) or it has no difficulty to convert data into the required format, users and translators are interested in working in interactive environments. The size of the documents under translation is another effective feature in encouraging translators to use TM. Naturally, when a text is large, it is more likely to find repetitive parts in it.

The areas of language whose texts useful for storing in TM and for subsequent translation with TM include wen pages, advertisements and marketing, communication and journalism, technology and law. For instance, we can mention user's manuals and booklets pertaining to repairing and maintenance of different machinery in technical areas, justifications and licenses in legal areas, the annual reports and contracts in the areas of commerce and commercial law, and updates and reviews in the areas of Internet and websites.

An interesting point about the TM applications is its ability to be merged with other methods. As an example, TM which is a corpus-based method can be combined with some statistical approaches. As TM can only translate the cases similar to those existing in its memory, some parts of the texts are left untranslated. In this case, the task of translating the remaining parts can be carried out using statistical approaches.

There are other factors affecting the degree of using TM by the users. The first one is the format of the texts a translator more often uses. Using TM seems more reasonable for a translator who does most of his/her works in electronic format rather than a translator who often works with pen and paper.

The second factor is the way of working – individual or team working. As we mentioned earlier, one of the advantages of using TM is keeping lexical consistency throughout the process of translating. When due to the time shortage or any other reason a team engages in translation, using TM can lead to a better output and prevent inconsistent lexical choices. Time factor is another criterion for using TM. The large volumes of texts requiring a rapid translation can also be exposed to a TM. Moreover in long texts the range of repetitious parts is higher than in the short texts. A TM can also be used as a part of automatic machine translation system. Statistical and example-based machine translation systems directly or indirectly use the linguistic knowledge existing in previous translations, So, TMs can combinationally be used in such systems.

An interesting usage of TM may be its application as an 'authoring memory' in writing environments with a controlled language for rewriting and reauthoring the written materials especially for technical texts. An authoring memory is a device which uses the previously written texts for rewriting the new versions of the same texts publishing frequently with little changes (Allen, J. 1991).

3. Methodology

In order to evaluate the performance of the TM software on the pairs of English and Persian languages, we tried to carry out an experiment using a TM system named "MetaTexis". MetaTexis for Word is a CAT **(C**omputer **A**ided **T**ranslation) tool. In the following lines we are trying to provide the readers with a short introduction about how a CAT tool like MetaTexis is designed to help translators translating texts, which has been quoted from its user's manual.

3. 1. MetaTexis as a CAT tool

CAT tools are designed to help translators translating texts. Translators do not translate words. For, even if words do have a meaning, they are not understood correctly unless they are interpreted within their context. Every translator has to learn that, rather than to translate words, he/she has to express the *meaning* of the source text with the words of the target language, using a new syntactical structure, leaving behind the structure of the source text.

In fact, a translator translates sentences. One could say that a sentence is both the smallest and the biggest unit a translator can handle. It is simply a matter of fact from a practical point of view. Translators usually do not translate whole texts, or whole paragraphs - even if they have to have them in mind as a background. Translators usually translate a text going from sentence to sentence simply because a sentence in general is the biggest text unit one can have a good overview about. At the same time, it is usually the smallest unit with a consistent meaning.

Unfortunately, text documents do not present sentences in a translator-friendly way. Text documents are optimized for reading, not for translating. The sentences are usually grouped together in paragraphs, and sometimes they cannot be easily distinguished.

And here is where a CAT tool starts to help, and we have arrived at the first basic function: **A** CAT tool presents sentences to the translator in a convenient way. Of course, CAT tools are not intelligent enough to *understand* the meaning of a text. Therefore, it is not guaranteed that it presents proper sentences in all cases. For this reason, it is a common habit to use the word "segment" rather than "sentence".

To present a segment is no big deal. There must be more in a CAT tool. The segments are not merely presented; they are presented in a way that you can enter the translation right below the source text. This enables a translator to compare source and translation directly without having to look at two different places. And, on top of this, the translation is stored in the same place as the source text so that you can come back to the source text at a later time to improve the translation. Thus, the second basic function of a CAT tool is to present a source segment and its translation as a unit. This unit is usually called a "translation unit", or "TU".

Even if a CAT tool would stop at this point, it would be very helpful for translators, simply because the process of translating is more efficiently organized. But a CAT tool can do much more. Especially in the case of technical translations and revised texts, a translator encounters segments which he/she has already translated before (or similar ones). In former times, when a translator realized that this was the case, he/she had to look up old translations, stored on paper in files. As this was usually very time-consuming, the translator very often decided, rather, to re-translate the segment from scratch. With a CAT tool, this is no longer necessary. A CAT tool provides functions which do this task for you.

Of course, a CAT tool cannot look up these segments in books or papers. There has to be a database where the source text and the translation, that is, the translation units, are stored. This database is usually called "translation memory", or "TM". Any CAT tool stores the translation units in a translation memory either immediately after each segment has been translated, or at a later time. The third basic function of a CAT tool is to store the translation units in a translation memory (TM) and to automatically look up the TM when a new segment has to be translated. Any result of the TM search is presented in a convenient way so that it can be re-used by the translator.

Through this feature, the working time for a translation can be drastically reduced, especially in the case of revisions or repetitive texts. (The re-use of translated segments is also called "leverage" or "leverage effect".). Of course, translators do not deal with sentences or segments as an atomic unit. Sentences are made of words. And to know the meaning of a sentence essentially depends on knowing what the individual words can mean. Therefore, before CAT tools were invented, dictionaries and glossaries used to be the

main tool of every translator. And looking up the relevant dictionaries was a time-consuming part of translators' working life. As with many other CAT tools, MetaTexis also includes special functions to make dictionary and glossary look-up more efficient. The fourth basic function of a CAT tool is the automatic look-up in terminology databases, and the automatic display and insertion of the search results.

MetaTexis is not a stand-alone-program. It runs in Microsoft Word. This means that all MetaTexis functions can be accessed through Microsoft Word. The great advantage of the integration in Word is that you do not have to learn a completely new program. You only have to learn some new functions. At the same time, all functions of Microsoft Word are available.

3. 2. The Experiment

As the very first step towards carrying out an experiment using an already designed system of TM like MetaTexis, we tried to compile and use a very large parallel corpus including pairs of equivalent sentences in English and Persian. As such a corpus was already available (Author, 2009), for the sake of this experiment we tried to improve its richness through adding some data. Moreover, a bilingual parallel corpus of chunks was also constructed using the main corpus, though, that is not very rich and complete yet. To localize the MetaTexis system to be suitable for the new pairs of English and Persian languages, the database of the system had to be created using the data from these two languages. For this reason, we imported the main corpus containing the parallel English and Persian sentences into the Translation Memory database of the system. Moreover, the corpus of the parallel chunks or segments was imported into the Terminology database of the system.

We started our experiment with a 100 randomly selected sentences from five various fields of study, namely, literature, religion, politics, technology and law. In fact, we wanted to examine the effectiveness of MetaTexis performance on the new pairs of languages in different types of texts. To what extent this TM system can automatically perform the translation task is the main goal of this experiment.

The material for this experiment consists of a test corpus including 100 sentences - 1461 words - derived from five text types each of which contains 20 sentences. The five single files are loaded into the system for translation separately, and the system launches start assistance sentence by sentence through the text.

`In this experiment, the MetaTexis TM system was used to assist translation of five different files belonging to different fields. The system translated the files separately sentence by sentence from English into Persian. Upon translating each sentence, there is a pause for user interaction with the system. That is, the system presents a translation suggestion for each sentence as a Translation Unit (TU), waits for the user interaction to correct some parts of it or approve it, and, does not go to the next Translation Unit until the user reacts in some way. Figure 1. represents the environment in which the first sentence has already been translated by the system and the user's revision on the suggested translation has already been applied. In this environment the second sentence has been shown while translating by the system. In the second Translation Unit the system first represents the segments belonging to the given sentence available in its Terminology database and then suggests a candidate translation of the given sentence, waiting for user's reaction.

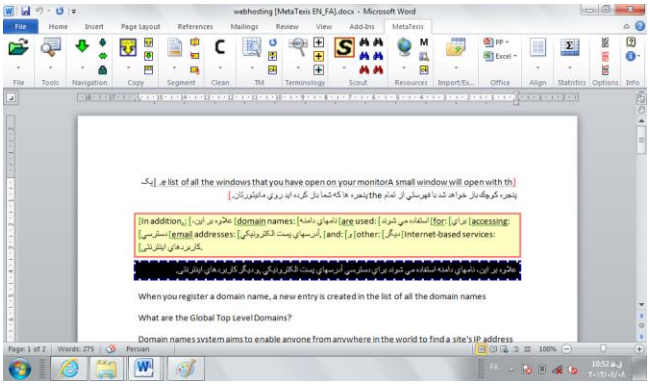

Figure 1. The environment of MetaTexis system while assisting the process of translation

4. Data Analysis and the Results

While translating each sentence by MetaTexis, there are some segments whose translation are not correct and need to be substituted by appropriate translations by the user. The incorrect translation units provided by the system to which we refer as translation errors have been divided into two categories. The first one deals with the errors concerning the incorrect use of words in translating segments. In this category fall the cases requiring deleting or inserting some words while revising the suggested translation. We call such errors "meaning errors". The second one deals with the incorrect word order of some part of a translation segment in Persian language which is called word order error. Consider the following two sentences. In the first sentence translated by MetaTexis and revised by a human translator as user, the translation error is a word order one, while in the second sentence the translation error is of the type of meaning error.

The rates of system errors corresponding to the five text types have been demonstrated in the following table:

Table 1. Rate of translation errors in different text types

Text type	Political	Law	Literary	Religious	Technical
Meaning errors (%)	0.074	0.054	0.138	0.053	0.026
Word order errors (%)	0.014	0.02	0.043	0.012	0.022
Total Rate of errors (%)	0.088	0.074	0.181	0.065	0.048

As Table 1. Shows, the highest rate of translation errors belongs to the literary texts in which the meaning errors are more prominent than word order errors. As the literary texts are the most difficult ones to be automatically translated, such results are rather expectable. Figure 2. may well be used to compare the translation error types across different text genres.

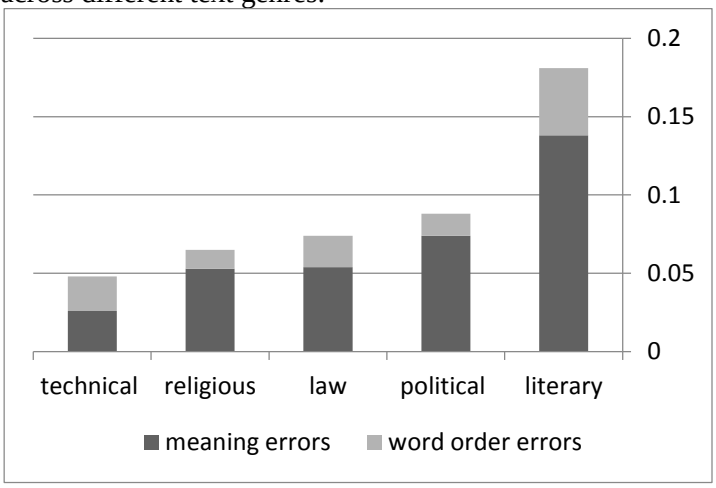

Figure 2. Rate of different translation errors in different text types

In this experiment the overall performance of MetaTexis system on English and Persian language pair has been calculated using "accuracy" metrics, and the results of the calculation of different translation error types for five various text types have been shown in Table 2.

Table 2. Total accuracy of MetaTexis based on the text type

Text type	Political	Law	Literary	Religious	Technical
Total accuracy (%)	91.1	92.5	81.8	93.4	95.1

Precision and Recall are two widely used statistical classifications. Precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search, and Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents (which should have been retrieved). As in this experiment, the number of

documents retrieved and the number of existing relevant documents are the same the total Accuracy of the system is to be calculated. In Figure 3. the comparison between different text genres can be made in more tangible way. As the figure demonstrates, the highest accuracy belongs to the technical texts in which there are usually more repetitive segments. As it was predicted, the literary texts are not very suitable texts to be translated with the help of a CAT tool like translation memory. In such texts there are less fixed segments or chunks to be stored in terminology database for reusing. Moreover, in literary texts more complicated structures as well as more figurative meanings tend to be used

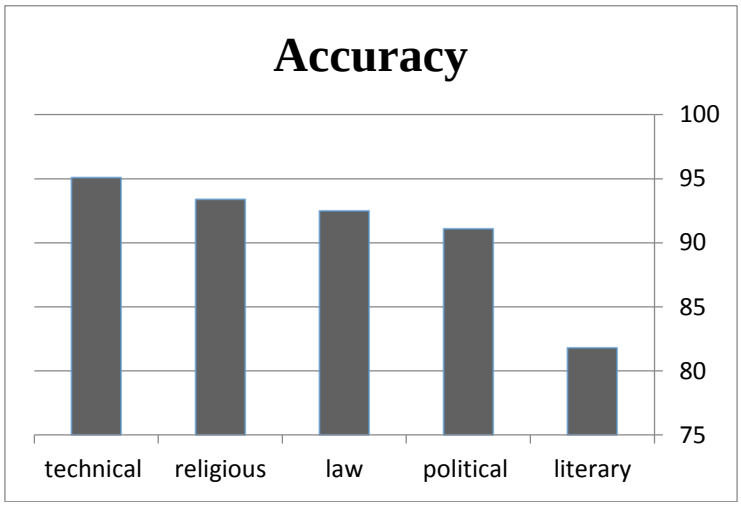

Figure 3. Total accuracy of MetaTexis based on the text type

The total accuracy of MetaTexis performance on English and Persian languages has been calculated as 90.78% which is very encouraging for the pair. There are, however, many ways to improve the performance of the system, among which the following cases can be mentioned. a) Adding some new functioned to MetaTexis system for entering lexical and grammatical rules. These language-specific rules may contain those for deleting the article "the" (which is an empty category in Persian) in the produced translations when it is additional. There may also be some rules for word order problem, infinitive verbs and their translation into Persian, different translations of prepositions accompanying verbs, and the like; b) improving the content of Terminology database (TDB) of the system using some heuristic statistical approaches; c) improving the content of translation Memory (TM) database of the system by enhancing the bilingual records of the main corpus, namely, English-Persian parallel corpus.

Conclusion and Further Development

In this study we tried to localize a Translation Memory system called MetaTexis to be applicable foe translation from English into Persian. The required adjustments on the system including importing suitable data to both TDB and TM databases of the system were made in order to carry out the experiment. The performance of the system was evaluated through an experiment using five different text types, including literary, law, political, religious and technical texts. The results of the experiment show that this Translation Memory system works well on technical texts in which fixed phrases as well repetitive segments are more prominent. The literary texts stand at the opposite side of technical texts. They show the highest rate of translation errors both on meaning errors and word order errors. The total accuracy of the system performance for the English and Persian language pair achieved 90.78 percent which is encouraging for the pair. As this is the first attempt towards localization of a Translation Memory system for Persian language, there are still many things to be done for improving the performance of the system in order to be more useful for translators involving Persian language. Enriching the internal databases of a TM system as well as adding some lexical and grammatical rules specific to the languages involved in the translation process are among the suggested procedures to be followed in subsequent studies in this respect.

References

Abney, S., 1991, Parsing by Chunks. In: R. C. Berwick (ed.), Principle-Based Parsing: *Computation and Psycholinguistics*, 257–78, Dordrecht: Kluwer.

Ahrenberg L. & Merkel M., 1996, On Translation Corpora and Translation Support Tools: A Project Report, Department of Computer & Information Science, Linköping University, Published in: Languages in Contrast, Papers from a Symposium on Text-bsed cross-linguistic studies, Eds. Aijmer, K., Altenberg, B. & Johansson, M., Lund University Press.

Allen J., 1999, Adapting the concept of "Translation Memory" to "Authoring memory" for controlled Language writing environment, at the 21st Conference of "Translating and the Computer", sponsored by ASLIB, 10-11 November 1999, London.

Arthern, P. J., 1981, Aids unlimited: the scope for machine aids in a large organization. *Aslib Proceedings*, 33, 309–319 (1981).

Freivalds, John (1999). The technology of translation. *Management Review*, Vol. 88, No. 7, pp. 48-52.

Gow, F. (2003). *Metrics for evaluating translation memory software*, MA thesis. University of Ottawa, Canada.

Kay, M. (1997). The proper place of man and machine in language translation. in *machine translation* volume 12, Nos. 1-2, 1997, 3-23 (reprint from 1980)

Kumar A. (2005). Design and Development of Translator's Workbench for English to Indian Languages, Translation Journal, vol. 9, No. 3, July 2005.

Lagoudaki E. (2006). Translation Memories Survey 2006, Imperial College London.

Melby, A. K., (1995). The Possibility of Language: A Discussion of the Nature of Language. John Benjamins, 1995, p. 225f.

Melamed, I. Dan., (1998). Word-to-Word Models of Translational Equivalence. Technical Report 98-08, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, USA.

Mosavi Miangah T. (2008) Translation memories and the translator. *International Journal of Translation*, Vol. 20, No. 1-2, PP. 97-106.

Mosavi Miangah T. (2009) Constructing a Large-Scale English-Persian Parallel Corpus, *Meta : journal des traducteurs / Meta: Translators' Journal*, Volume 54, Number 1, January 2009, p. 181-188.

Planas E. (2000) Extending Translation Memories, NTT Cyber Solutions Laboratories, Japan.

Schäler, R. (2008). Linguistic resources and localization. In: *Topics in language resources for translation and localization*. Edited by Yuste, Rodrigo, E., Benjamins Translation Library, pp: 195-214.

Simard M., Langlais P. (2001). Sub-sentential Exploitation of Translation Memories, *Proceedings of MT Summit VIII: Machine Translation in the Information Age*, Santiago de Compostela, Spain, 18-22 September 2001; pp.335-339.

Yamada, M, (2011). The effect of translation memory databases on productivity. From: *Translation Research Projects 3*, ed. Anthony Pym, Tarragona: Intercultural Studies Group, 2011. pp. 63-73. http://isg.urv.es/publicity/isg/publications/trp_3_2011/index.htm.

Wu, D. (1997). Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–404, September.