Introducing the Autshumato Integrated Translation Environment

Hendrik J. Groenewald
Centre for Text Technology (CTexT)
North-West University
Potchefstroom, South Africa
handre.groenewald@nwu.ac.za

Wildrich Fourie
Centre for Text Technology (CTexT)
North-West University
Potchefstroom, South Africa
wildrich.fourie@nwu.ac.za

Abstract

Translation is an indispensable process for socioeconomic and cultural development in a multilingual society. The use of translation tools such as translation memories and machine translation systems can be beneficial in supporting the human translator. Unfortunately, the availability of these tools is limited for resource-scarce languages. In this paper, we introduce the *Autshumato Integrated Translation Environment* that facilitates a comprehensive set of translation tools, including machine translation and translation memory. The *Autshumato Integrated Translation Environment* is specifically developed for translation between the official South African languages, but it is in essence language-independent and can therefore be used to translate between any language pair.

1 Introduction

South Africa is a culturally diverse country, with a large number of different ethnic groups. This rich cultural diversity is the main reason for South Africa having no fewer than eleven official languages. The South African Constitution (Republic of South Africa, 1996) guarantees equal status to each of the eleven official languages.

Translating between these eleven languages is an immense task, and unsurprisingly, English often acts as the primary language for government communication. The problem with English being the lingua franca is that the other 10 official languages are marginalised. An example of this is the fact that the South African government does not have the capacity to produce parliamentary records in all eleven official languages. Multilingual parliamentary records are one of the most important resources for machine translation (MT) projects elsewhere in the world. One such a project is the *EuroMatrix* project, a statistical and hybrid machine translation project for translating between European languages (see www.euromatrix.net). The *EuroMatrix* project uses the parliamentary records of the European Parliament, available in all 23 official languages of the European Union, as one of their main resources for the development of machine translation systems. The parliamentary records of the South African government are only recorded in English on both provincial and national level. The result of this is that a large number of South African citizens are denied access to important government information in their home language, since English is the mother tongue of only 8.2% of the South African population (Van der Merwe and Van der Merwe, 2006). Another negative aspect is that researchers and developers working on the development of machine translation systems for South African languages do not have access to as large amounts of data as their European colleagues.

The South African Government is aware of the importance of elevating the status and promoting the equal use of all eleven official languages on all levels. As a result of this, the government is involved in a large number of initiatives and projects to promote multilingualism. One such as project is the *Autshumato*[1] Project that was

[1] Autshumato was a Khoi-khoi leader that worked as an interpreter between the Europeans and the Khoi-khoi people during the establishment of the Dutch settlement at the Cape of Good Hope in the 17th century (Giliomee and Mbenga, 2007). Autshumato can for this reason be viewed as one of the first translators in South Africa.

Proceedings of the 13th Annual Conference of the EAMT, pages 190–196,
Barcelona, May 2009

commissioned by the South African Department of Arts and Culture for the development of translation tools and resources for the eleven official South African languages. An integral part of the *Autshumato* Project is the *Autshumato Integrated Translation Environment* (ITE), which is the subject of this paper.

The rest of this article is organised as follows: the next section provides information about the end-user requirements for the *Autshumato* ITE; Section 3 describes related work; and Section 4 provides detailed information about the design and implementation of the system. The main functionalities are discussed in Section 5, while the paper concludes with a discussion of future work in Section 6.

2 End-user requirements

As indicated in the previous section, the South African government's translation agencies do not have the capacity to translate all government documents and communications into all of the official languages. The magnitude of this problem is increased by the lack of availability of translation tools for the eleven official languages. The purpose of the *Autshumato* project is to develop tools and resources that will help government translators increase both the quantity and the quality of their translation work. Although these tools are specifically developed for use by government translators, it will in future be released under an open source license for use by all translators.

Since government translators are the most important clients of the *Autshumato* Project, one of the first objectives of this project was to determine the end-user requirements of the translators working at various government departments. The end-user requirements were elicited by means of questionnaires and joint-requirements planning (JRP) sessions. The joint requirements planning sessions were held with translators working at the offices of the South African National Language Service (NLS) and members of the development team. The results of the questionnaires and joint requirements planning sessions can be summarised as follows:

- Inaccurate translations – The overload of work that translators receive has a negative influence on the quality of translations.
- A system that contains a memory of terms previously coined and used in similar documents is not available. The NLS has a terminology-database, but this is not electronically accessible by the translators due to proprietary licensing restrictions. The translators do however receive a hard copy of the terminology database.
- In the past, some translators used SDL Trados (see www.trados.com), a translation memory (TM) system. SDL Trados is not used anymore, due to software compatibility and licensing issues.
- No online machine translation systems exist for South African languages. In general, insufficient online resources are available for African languages.
- Translators do not use a standard file naming convention. The consequence of this is difficulty in finding parallel translations of the same document.
- Hardware – Most translators do not have dual monitors connected to their computers. Dual monitors can be useful when translating between electronic documents.
- Translators make no use of previously translated documents. Some documents contain duplicate information that can be copied instead of translated again. Previously translated documents can also act as a template for new documents on the same topic or genre.
- Some translators perform translation by overtyping on the original document. They do not keep copies of the original documents, making it impossible to obtain parallel-translated documents.
- The South African government have decided to implement open source software in all government departments. Proprietary operating systems and word processors are being phased out in favour of open source solutions. The problem with this is that the change to open source is happening very slowly. Some departments have already implemented open source, while others are still using proprietary software. The implication of this is that all software developed for the

South African government must be open source and cross-platform compatible.

- Most translators use Microsoft® Office Word as their default translation environment. They are used to the graphic user interface of Microsoft® Office Word, and therefore would prefer a computer assisted translation program with a similar user interface to that of Microsoft® Office Word.
- Translators working at government departments have basic computer skills. Translators require computer assisted translation tools with a shallow learning curve that are simple and intuitive to use.
- Surprisingly, most translators are not opposed to machine translation systems.

The abovementioned preferences were analysed and rendered into end-user requirements, which steered the design and development of *Autshumato* ITE.

3 Related Work

Translation memories and machine translation systems are traditionally seen as opposing technologies. Although combining translation memory with machine translation is not a novel idea (Mügge, 2001 and Shuttleworth, 2002), very few computer-assisted translation tools exist that incorporate both technologies. We have an active interest in machine translation, and believe that both machine translation and translation memory have an important role to play in the translation process, especially in cases of languages with limited resources. We therefore decided to incorporate both technologies in the *Autshumato* ITE.

Other computer-assisted translation (CAT) applications that employ both machine translation and translation memory include the SDL Knowledge-based Translation System™ (http://www.sdl.com), Lingo24 ContexTrans (www.lingo24.com) and the ESTeam Translator© (www.esteam.se).

4 Design and Implementation

4.1 Integrated Solution

The project requirements (see Section 2) indicate the need for an easy and effective integrated translation environment for translators with basic computer skills. The term integrated refers to the fact that all the translation tools/resources required by the translator are accessible from within a single environment. These translation tools and resources include translation memory, machine translation, and term banks (glossaries). A diagram of the tools/resources that provide input to the ITE is displayed in Figure 1. The ITE must furthermore support open standards (i.e. TMX, XLIFF, etc.) to ensure compatibility with other translation tools. Using open standards also ensures that information is always accessible, and never becomes locked away within a proprietary format requiring a legacy application to access.

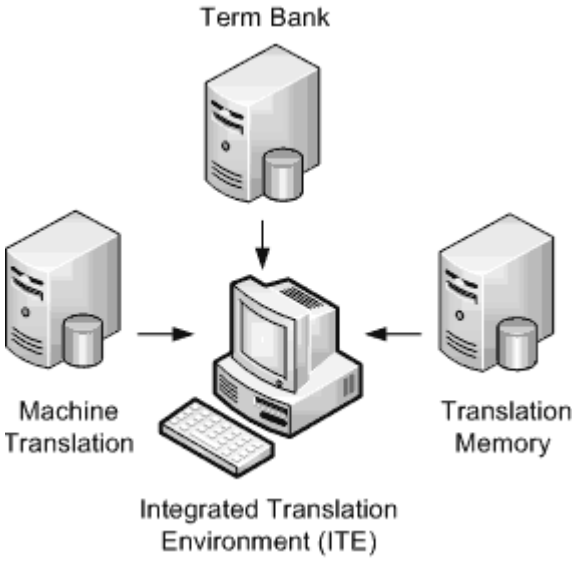

Figure 1. Translation Tools incorporated in the ITE.

4.2 Translation Workflow

Another important requirement is that ITE must be streamlined to provide a simple linear translation workflow. By following the linear flow: *Open* → *View* → *Translate* → *Edit* → *Restart*, we ensure that the translation of a document proceeds along a logical path. A diagram of the linear translation workflow is displayed in Figure 2.

4.3 Open Source Components

Since *Autshumato* is an open source project, we are utilising various existing open source components in the ITE. The main components used are the OpenOffice.org office suite, the OmegaT® CAT tool and the Moses Statistical Machine Translation system.

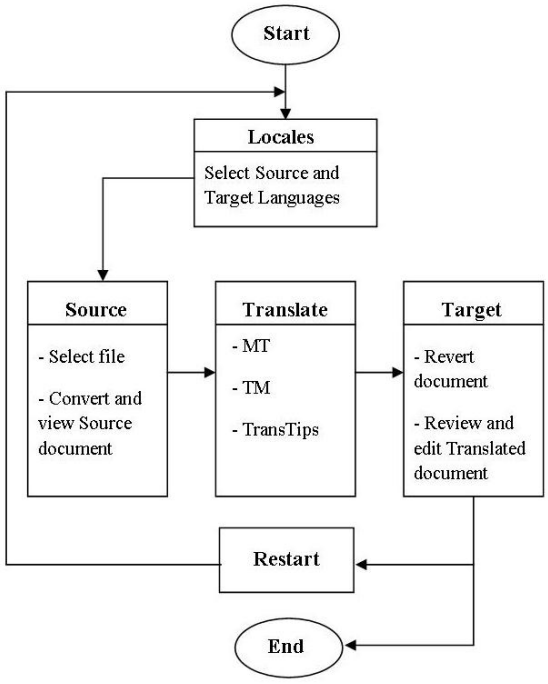

Figure 2. Workflow diagram.

OpenOffice.org is a popular office suite used for word processing, databases, spreadsheets and more. It can open documents from various other office packages and works on all the main operating systems and computers. It delivers documents in the international open standard and is open-source, which is free to everybody (www.openoffice.org). We use OpenOffice.org for its open document standard and relative ease in incorporating into other programs. Another important motivation for using OpenOffice.org is its resemblance to Microsoft® Office, since the translators indicated their preference for a translation environment with a similar interface to and all the functionalities of Microsoft® Office Word, during the end-user requirements elicitation.

OmegaT® is a popular CAT tool, which enables translators to effectively utilise TM, fuzzy matching and term banks to assist in the translation process (www.omegat.org). It already incorporates a large number of the functionalities required for *Autshumato* ITE, and is a very active open source project.

Moses is a statistical machine translation system using factored phrase-based beam-search decoding to deliver in-time translations. It has to be trained for every language pair with a collection of parallel corpora (Koehn et al, 2007). The Moses statistical machine translation system was chosen because it is currently one of the most advanced open source statistical machine translation systems.

The ITE is being developed with the Java programming language, a popular programming language for open source projects. Java was chosen to ensure compatibility and easy integration with other open source projects. The ITE is being primarily designed specifically for translators translating between the eleven South African languages; it can however be easily adapted for translating between any language pair in the world.

4.4 User Interface

In order for the ITE to appear simple and intuitive to use, the user interface should be logical and foreseeable. It is imperative that users do not get confused with endless options and functions. Functions that are more frequently used (like TM, MT, and TransTips) are automated to provide simpler interaction. The rest of the most commonly used options and functions are present on the toolbar. Usability is also improved by specifying the highlight colours for the source and translated texts, making it easier to identify untranslated sections.

The original design of the graphic user interface (GUI) entailed a split screen approach. This approach caused the GUI to be vertically split into two equally sized parts, with the source text being displayed on the left and the target text on the right. Translators disapproved of the split screen idea; they were concerned that the split screen would result in smaller working areas, which would in turn strain their eyes. We agreed that the split screen would not be practical, especially when working with an older desktop computer or notebook with a small monitor. We decided that a better alternative to the split screen approach would be to display the source and the target texts in separate windows, with the actual translation taking place in a third window.

The three windows are respectively named “Source”, “Translate” and “Target”. The three windows conform to the “View”, “Translate” and “Edit” phases in the unidirectional translation workflow. More information on the three windows is provided in the next section.

5 Main Functionalities

5.1 Source Window

The source window displays a read-only copy of the source document in its original form. The

translator is prevented from editing the source document to ensure true parallel target translations in the event of a single source document being translated into more than one target language. The source document serves only as a reference for comparing the target document during the translation process. The source window contains an embedded OpenOffice.org Writer for displaying the source document.

The source document is automatically converted to OpenOffice.org Writer format, without losing any text, pictures, tables, graphs etc. The source window supports the following file formats: Microsoft® Office Word (*.doc), OpenOffice.org Writer (*.odt), Xml (*.xml) and Html (*.html). Other formats like OpenDocument Spreadsheet (*.ods), OpenDocument Presentation (*.odp), and Office Open XML are to be included in future versions. Figure 2 shows an example of a document being displayed in the source window.

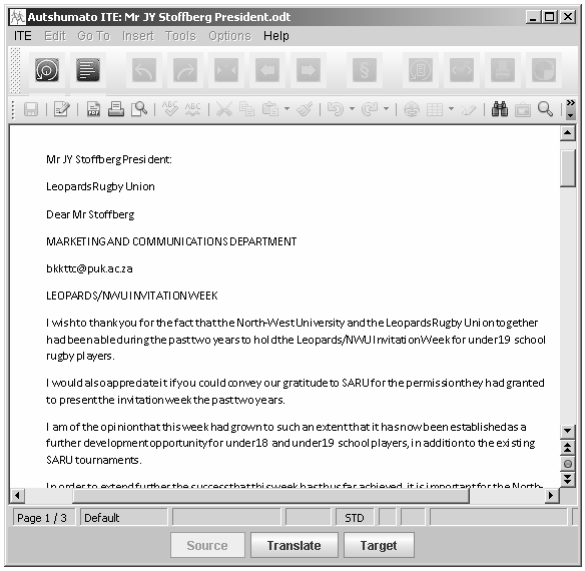

Figure 2. The Source window.

5.2 Translation Window

The Translation Window is where the actual translation is performed and consists of an embedded (and modified) OmegaT® CAT tool. The window is split into three panels: Translate, Fuzzy Matches and Machine Translation. A screenshot of the translate window is showed in Figure 3.

The translate panel displays segments of the source document for translation. Two choices of segmentation, namely paragraph and sentence segmentation are available to the user. Every segment contains a space where the translation of the involved segment must be inserted. The translator translates the entire source document in a segment-by-segment fashion in the translation window.

Various procedures work in the background to support the translator in creating an accurate translation. When a segment is activated, the program searches the TM for possible matches and displays the five best matches in the fuzzy match panel. The segments are matched by determining the Levenshtein distance (Levenshtein, 1965) between the sentence or paragraph in the involved segment and the translations contained in TM. All newly translated segments are constantly included in the TM, which greatly speeds up the translation of documents in the same context or domain.

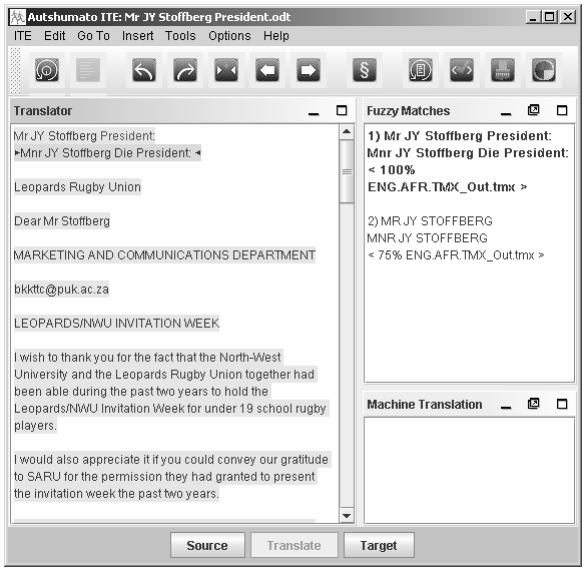

Figure 3. The Translation window.

Every word in the segment is matched against the glossary for finding word translations. Words for which matches are found are displayed in a different colour than the rest of the text in the segment. If the translator hovers the cursor over a matched word, the translation of the particular word is displayed in a pop-up window. These word translations are called TransTips. Figure 4 shows an example of a TransTip displaying the word “uitnodiging”, which is the Afrikaans translation of “invitation”. The applicable TM and glossary are loaded according to the source and target languages specified at the beginning of the translation workflow.

Upon entering the translation window, the entire source document is submitted to a server

running the Moses machine translation system. After the document has been translated by the server, it is downloaded to the ITE. The machine translation of every segment is displayed in the Machine Translation panel.

The translator can specify the editing behaviour of the segments. This provides the options of leaving the translate segment empty, automatically inserting the best fuzzy match, automatically inserting the machine translation, or simply copying the source text for overtyping. The translator can also combine fuzzy matches and machine translations by copying translated phrases from the Fuzzy Matches and MT panels.

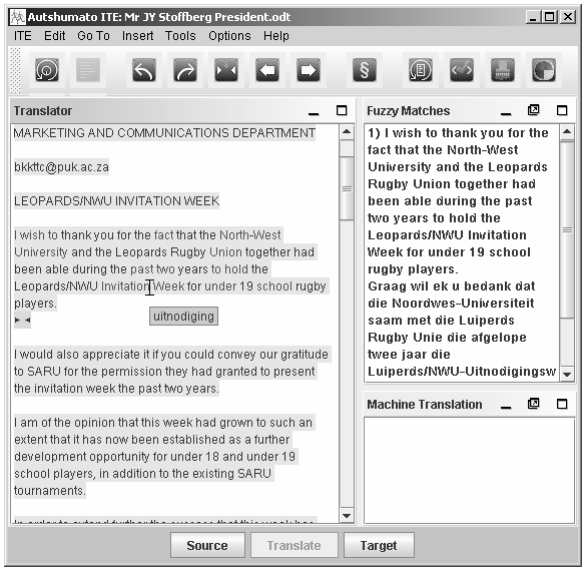

Figure 4. A TransTip being displayed.

5.3 Target Window

The target window displays the translated document in another embedded OpenOffice.org Writer component. It is fully editable and contains most of the OpenOffice.org Writer word processing functionalities.

Translators working for Government departments are required to produce translated documents with the same formatting as the original source documents. For this reason, the text formatting (style, font, size, colour, etc.) of the source document is saved in a skeleton file before the document is imported into the ITE. This formatting is automatically applied to the newly translated document to ensure it looks exactly the same as the original.

In cases where a word (or phrase) in a source sentence contains formatting different from the rest of the sentence, it is not always possible for the ITE to automatically apply the formatting to the correct target word, especially when the target word is not contained in the glossary. To overcome this problem we display tags in the source text to indicate formatting in the translation window. The tags usually appear in pairs, with rare occurrences of singular tags. The translator is required to carry over the correct formatting to the target text by using the Tag Painter tool. The ITE warns the translator when a segment contains source text formatting that has not been carried over to the target text. The advantage of the Tag Painter Tool is that the translator spends less time formatting and more time on translation.

This translated document in the target window can be compared to the original document in the source window. The translator is forced to save his/her work using a standard file naming convention, this is to prevent instances where the source document and the translated document cannot be linked. The target document can be saved in any of the following output formats, OpenDocument text (*.odt), Microsoft® Office Word (*.doc) or Portable Document Format (*.pdf). Figure 5 shows the translated document in the target window.

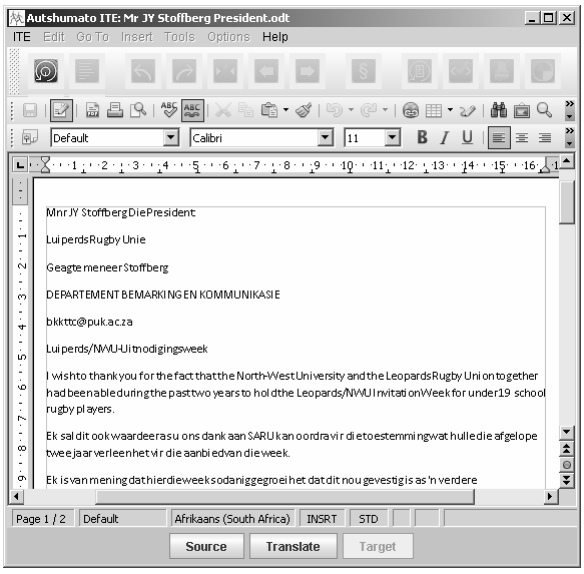

Figure 5. The Target window.

6 Conclusion

In this paper, we have introduced *Autshumato* ITE, a free and open source integrated translation environment that facilitates a variety of translation tools and resources. An important

feature of *Autshumato* ITE is that it offers both machine translation and translation memory to help translators to improve the quality and quantity of their translations.

Future work includes extending *Autshumato* ITE's capability to incorporate a fully-fledged terminology management system that will be accessible to both translators and terminologists. We are also interested in incorporating a document management system. The ITE currently supports only a limited number of file formats and we want to extend the list of supported file formats for input and output considerably .

We have vast experience in the development of spelling checkers for South African languages and want to apply this knowledge to create open source spelling checkers that can be implemented into the ITE. Other functionalities and improvements will be made based on the feedback we expect to receive from the users of *Autshumato* ITE.

We are also creating a website that will host an open source community for the *Autshumato* project. The website will contain downloadable resources, source code and complete binary packages for deployment.

One of the biggest challenges we are facing, is the development of machine translation systems for the official South African languages. Machine translation systems require large sets of parallel corpora and as previously mentioned in this article, this is a very scarce resource for South African languages. We are in the process of gathering parallel data for developing and improving existing machine translation systems for South African languages. We are aware of the fact that statistical machine translation might not be the ideal approach for creating machine translation systems for resource-scarce languages with small amounts of parallel corpora. For this reason where are doing research on alternative ways than mere addition of parallel data, to improve the output of the statistical machine translation systems. We do however believe that *Autshumato* ITE will contribute in generating more parallel data by stimulating translation to resource-scarce languages. In turn, this would help us to improve the quality of statistical machine translations for these languages.

References

Giliomee, Herman and Mbenga, Bernard. 2007. *New History of South Africa*, Tafelberg, Cape Town.

Koehn, Philipp., Hoang, Hieu., Birch, Alexandra., Callison-Burch, Chris., Federico, Marcello., Bertoldi, Nicola., Cowan, Brooke., Shen, Wade., Moran, Christine., Zens, Richard., Dyer, Chris., Bojar, Ondrej., Constantin, Alexandra., and Herbst, Evan. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demonstration and Poster Session*. Prague, Czech Republic. 177-180

Levenshtein, Vladimir I. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10:707-710.

Mügge, Uwe. 2001. The Best of Two Worlds - Integrating Machine Translation into Translation Memory Systems: A universal approach based on the TMX standard. *Language International*, 13 (6): 26-29.

Republic of South Africa, 1996. *Constitution of the Republic of South Africa,* Act 108 of 1996. Government Printer, Pretoria.

Shuttleworth, Mark. 2002. Combining MT and TM on a Technology-oriented Translation Masters: Aims and Perspectives. Proceedings of the *6th BCS EAMT Workshop Teaching Machine Translation*. Manchester, England.

Van der Merwe, I.J. and Van der Merwe, J.H. 2006. *Linguistics Atlas of South Africa*, SUN PReSS, Stellenbosch.