The Temple Web Translator

Rémi Zajac and Mark Casper
Computing Research Laboratory
New Mexico State University, Las Cruces, NM 88003
{zajac,mcasper}@crl.nmsu.edu
Electronic Copy: http://crl.nmsu.edu/temple/twt/twt.aaai97.html

Abstract

New Web sites in foreign languages are appearing everyday, and language barriers threaten to atomize the World Wide Web into closed linguistic communities. The Temple project has developed an open multilingual architecture and software support for rapid development of machine translation systems for assimilation purposes. The targeted languages are those for which natural language processing and human resources are scarce or difficult to obtain. The goal is to support rapid development of machine translation functionalities in a very short time with limited resources. A Web front-end allows a user to submit an HTML page to the machine translation server and get back the translated page in English.

Introduction

Web Machine Translation (MT) is already a reality for several millions of Japanese Web surfers who prefer to read an English HTML page in bad Japanese rather than in the English original. These crude MT systems (see for example Church & Hovy 93, Kay 80) have several common characteristics:

- They are small enough to run on low-end PCs,
- They are robust enough to process any HTML input document,
- They are fast enough so as to add only a small delay in accessing a document,
- Despite the low quality of the translation they have wide acceptance, a typical feature of MT for assimilation as opposed to dissemination purposes,[1]
- They are cheap.

A *multilingual* Web translator would limit the advantage of small size, making personal multilingual MT systems impractical for more than a few languages. When true multilinguality is concerned, running an MT system on a workstation is probably the adequate answer, although a small size for each component is still desirable. The best state-of-the-art MT systems produce much higher quality translations than those of cheap Web translators. After all, they are mostly targeted for *dissemination*, whose quality requirements are far more stringent than for assimilation. However, to use these traditional systems for assimilation is probably an overkill and could even be counter-productive in the long run; these systems are usually specialized for some specific texts and domains and are rather difficult to maintain and upgrade.

The Temple project at CRL[2] demonstrated the feasibility of rapid development of multilingual MT systems for assimilation purposes by using a selection of state-of-the-art MT technologies, e.g., finite-state technology and automated acquisition of languages resources. This article gives a brief overview of the Temple project, a description of the Temple Web front-end, and discusses aspects of MT for the World-Wide Web, some of which are being researched in the on-going Corelli project at CRL.

Background: the Temple Project

The Temple project has developed an open multilingual architecture and software support for rapid development of extensible Machine Translation functionalities. The targeted languages are those for which natural language processing and human resources are scarce or difficult to obtain. The goal was to support rapid development of machine translation functionalities in a very short time with limited resources.

Glossary-Based Machine-Translation (GBMT) is used to provide an English gloss of a foreign document. A GBMT system uses a bilingual phrasal dictionary (glossary) to produce a phrase-by-phrase translation. Translation (based on phrase pattern-matching) is fast and accurate regarding the content of the document and browsed documents can be translated almost in real-time. A GBMT system for a language pair is also extremely simple, cheap and fast to develop. Moreover, all language resources used by the system are entirely under the control of the user.

The Temple GBMT system has been integrated in the Temple multilingual analyst's workstation (Zajac & Vanni 96). This workstation uses a Tipster document server developed at CRL (Grishman 95) to store and retrieve

1. Assimilation is what a Web surfer or an analyst does by browsing a large number of documents for retrieving some information. Dissemination is, for example, the publication of a manual or a commercial brochure.

2. http://crl.nmsu.edu/temple.

documents and associated information, such as linguistic structures produced by some NLP component. The analyst workstation offers to an English-speaking analyst a variety of tools to browse sets of documents in Arabic, Japanese, Spanish and Russian, including a Unicode-based (Unicode 96) multilingual editor, a simple machine translation functionality that is implemented using the GBMT system and other information retrieval tools (Callan et al. 92).

Glossary-Based MT

GBMT systems put an emphasis on a feature of natural language that is frequently overlooked by (computational) linguists, i.e., the fact that the meaning of a sentence is very often non-compositional. Idioms, collocations and specialized terminology, which occur much more often than is usually assumed in real texts, must all be listed individually in the lexical database of the system. This is the very reason why older systems, whose dictionaries and other lexical resources have frequently been developed over several decades, still out-perform newer systems using better technology.

GBMT, as exemplified by the Temple system, is a suitable paradigm for MT for assimilation. Using finite-state technology throughout, a GBMT system can be both fast and compact. By using various levels of defaults in case of failure at a higher level, translation is also very robust. Moreover, these types of systems are much easier to develop than more traditional MT systems; all the basic resources used by the system can be presented to a user in a very intuitive and declarative way, greatly facilitating the maintenance and evolution of the system.

Developing lexical resources for MT systems is the highest-cost component for *any* type of system. Automating the acquisition of such resources is therefore a very important topic, addressed in current MT research under the learning or induction paradigms. The restriction to finite-state technology should bring important advantages for learning since induction algorithms for regular languages give much better results than for other classes of languages.

Glossary-Based Machine Translation (GBMT) was first developed at Carnegie Mellon University as part of the Pangloss project (Cohen et al. 93, Nirenburg et al. 93, Frederking et al. 93), and a sizeable Spanish-English GBMT system was implemented. The Temple project has built upon this experience and extended the GBMT approach to other languages: Japanese, Arabic, and Russian. This experience with other languages has provided significant insights for the development of a versatile GBMT engine and for the use of off-the-shelf components for building a complete MT system. Building a generic platform for integrating various MT systems in a single flexible user environment, built upon the Tipster document architecture, has also been a valuable experience for developing generic natural language processing support systems.

The User Perspective

The Temple Analyst's Workstation is incorporated into a Tipster document management architecture and allows both translator/analysts and monolingual analysts to use the MT function for assessing the relevance of a translated document or otherwise using its information in the performance of other types of information processing. Translators can also use its output as a rough draft from which to begin the process of producing a translation, following up with specific post-editing functions. The user (translator or analyst) can:

- Browse a collection of documents managed by a Tipster Document Manager using a Collection/ Document browser,
- View and edit foreign language documents or their English translations using the multilingual Tipster Editor for Documents,
- Translate foreign documents using the generic translation function,
- Browse and edit lexical resources (bilingual dictionaries and glossaries) using the multilingual Temple Lexical Editor.

Although the translation provided by the system is only a phrase-for-phrase (or word-for-word) gloss of the original, the system is entirely under the control of the user who can modify any essential part of it, i.e., the dictionaries and glossaries. From a user's point of view, the system is predictable, responsive, affordable and easy to use and maintain. Like advanced MT systems, it uses reliable morphological processors and taggers, components which are relatively inexpensive, require little or no maintenance, and greatly enhance output quality of GBMT.

The MT Developer Perspective

A Multilingual Architecture. The Temple architecture uses a multilingual text library developed at CRL to support multilingual text processing. This library, available for Unix systems, is capable of handling a large number of character codesets and provides multilingual string processing functionalities and character code conversion for a large variety of codesets. A multilingual Motif text widget that can be embedded in higher-level applications (as in the Temple Lexical Editor) and a simple multilingual text editor are also supported. This library proved to be a major asset for the project since few comparable functionalities for the range of languages processed in the Temple project are available in the Unix environment.

Although full Unicode support was a goal of the project, this could not be achieved entirely. Currently, only the Arabic morphological analyzer has built-in Unicode support (as well as other codesets); however, a full Unicode library should be available for use in the Corelli project (see below).

Reuse of Machine-Readable Dictionaries.

Bilingual dictionaries are processed versions of various machine-readable dictionaries (MRDs), for example, the Collins Spanish-English dictionary (or any other MT dictionaries that have been restructured to conform to Temple's lexical format, see Stein et al. 93). Since morphological analyzers and dictionaries may come from different sources, they may have incompatible lexical representation, as it happened for the Japanese-English dictionary and the Japanese morphological analyzer. In such cases, integration is achieved by mapping the dictionary, including, for example, part-of-speech information, to a standardized format, and by developing a filter that maps the morphological analyzer results to that structure.

Reuse of Natural Language Processing Components.

An important decision in the Temple project was to use available NLP components and resources whenever possible (Farwell 94, Penman 88, Matsumoto 93). This led to the definition of an open architecture that provides support for integrating external tools. Some Temple components have been developed as part of the project (e.g., the Arabic and Russian morphological analyzers) but have been integrated using the same methodology as other tools. The Temple architecture is built around a uniform internal canonical linguistic representation for all languages; all components read or map their results to this representation, which uses Tipster annotations. All NLP tools are encapsulated using Tcl wrappers for mapping the tool representation to the Temple representation.

Morphological information is transferred from the source to the target language using morphological transfer tables that map categories and features from a source lexical item to the equivalent English lexical item. The GBMT engine itself is fully generic and is parametrized by a bilingual glossary and a morphological transfer table.

Semi-automatic Development of Glossaries.

Small glossaries (between 2,000 and 20,000 entries depending on the language) have been developed for each language. The acquisition process is as follows:

1. An Ngram extraction program is used to collect recurrent word patterns in a given corpus (Davis et al. 95);
2. This set of patterns is loaded in the Lexical Database as partial glossary entries;
3. The translation of each entry is added manually.

The development cost of such glossaries remains relatively low since the structure and the information encoded in a glossary entry is very simple.

MT for the Web

Although the Temple system was not originally designed for Web browsing, it was nevertheless designed for analysts browsing large numbers of relatively small, a few pages at most, heterogeneous documents, in different domains, a variety of styles, and different languages. The characteristics of the texts are quite close to most of what is found in Web sites, and the analyst task is also close to the typical behavior of a Web surfer. To test the adequacy of this hypothesis, we developed a Web front-end for the Temple MT system which includes an HTML parser. This experiment is described hereafter in the following section.

The Temple system is a laboratory prototype which was used to demonstrate the feasibility of the approach on a realistic scale. In a second phase, which started during summer 1996, the Corelli project is re-implementing some key components of the system and developing the Temple technology in the following directions:

- Developing a new NLP architecture to serve as the integration layer of a multi-engine MT system (in cooperation with the Mikrokosmos project).
- Developing and extending the pattern-matching technology that is at the core of the GBMT system.
- Developing a comprehensive set of lexical acquisition tools for the automated development of glossaries and other lexical resources.
- Developing a Translation Editor integrated in a Document Preparation System (FrameMaker).
- Applying this technology to new language pairs.

The Temple Web Translator

The Web surfer is not interested in MT, only in the document's content, and any impediment to straightforward access is considered detrimental. Therefore, the MT system should be as unobtrusive as possible. In the ideal scenario, both browsers and servers would support relevant features of the HTTP protocol, i.e., language negotiation (only a few currently provide this functionality), and the charset and language attributes. These features are used in building applications that automatically select the appropriate language and character set for display.

These features however do not help to process a document mixing several languages (e.g., for many Japanese Web pages) and character sets, something which is unavoidable when, for example, an unknown foreign word needs to be inserted in the English translation. The

current HTML standard does not provide support for multilingual documents, although the World-Wide Web Consortium (W3C) has issued a draft on the internationalization of HTML (the so-called 'i18n' draft, see Nicol et al. 96) and proposes that a future version of HTML support Unicode and a 'lang' attribute (see also Carrasco Benitez 96 for an overview of issues related to i18n and multilingualism).

In the ideal scenario, on the user side, a user would set an ordered list of preferred languages for browsing and get back the document in one of the languages requested. On the server side, the server would automatically select the appropriate version if available at the site. If the document is not available in any of the languages specified by the user, there are two possibilities for translating a document in a requested language since the MT system could be located at the server or at the browser. If the MT system is located at the server site, the server could send the MT system a document in one of the languages accepted by the system and get a translation in one of the languages requested by the user. If the MT system is located at the user site, language negotiation can be used by the browser for requesting a translation before displaying the document.

There are also classical problems in relating and displaying a document and its translation. The source and its translation could be displayed side-by-side in different browsers, side-by-side in the same browser, or using an interlinear model with sentence level translations occurring adjacent to corresponding source sentences. Several proposals have already been made to extend HTPP and HTML to handle this set of problems (e.g., Bryan 96).[3]

While waiting for full support of the i18n draft, we have implemented an interface in which a user can send a translation request to the Temple Translation Server: the user provides the URL and the Translation Server sends back the English translation of the foreign Web page in a new browser.

The Temple Web Translation architecture is pictured in Figure 1. A Web browser displaying the Javascript-enabled Web Translator Tool (Figure 2) allows the user to enter the URL of a document to be translated. The tool generates a translation request for the indicated document, which is delivered to CRL's Temple Translation Server via CRL's Web server and a CGI script. The Temple Translation Server retrieves the document from the Web, parses the document structure, translates its textual contents, and returns the translated document, with original HTML formatting intact, to the user. The newly translated document is then displayed in a separate browser window.

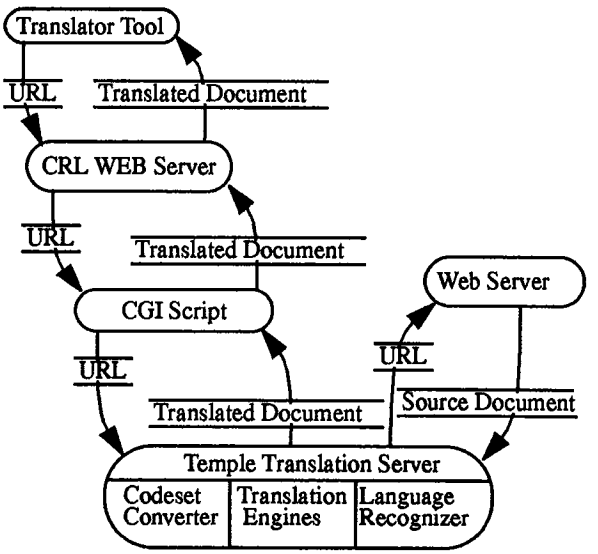

Figure 1: CRL Web Translation System Architecture.

The current implementation of the Web Translation System also supports delivery of a translated document to the user via e-mail. In the future, we will also provide an e-mail interface to the Translation System to allow for asynchronous access: as currently envisioned, the user will be able to submit either a URL or a mime-encoded document for translation.

For translating Web pages from any site in the world, several problems need to be addressed. After retrieving the selected document from the Web, the Translation Server is faced with the tasks of language and codeset determination, possible codeset conversion, and HTML parsing.

First, the server must determine the document's language and codeset characteristics. Our first Web Translation System implementation assumes that the HTTP server delivering the document correctly returns the document's codeset in the HTTP header. If the codeset of the document can not be determined or does not appear to match one of the supported translation languages, a corresponding error is returned to the user. If the user is confident that the document is indeed in one of the supported languages, he/she may specify the correct language and codeset via the Translator Tool and resubmit the request.

3. All references on internationalization of the Web can be obtained at the W3C web site, http://www.w3.org.

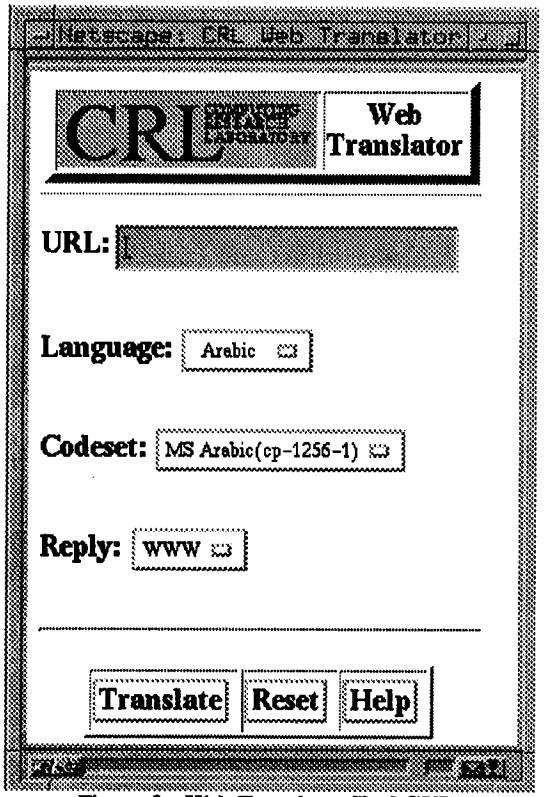

Figure 2: Web Translator Tool GUI.

Once the codeset of the document is known, it will be possible to determine the associated language for many, if not all, languages. For some documents, the language mode will be included in the HTTP header by the Web server. When this is not the case, many documents will be encoded using codesets that are uniquely associated with a language. The remaining (difficult) documents then will be those encoded using codesets such as ISO-8859-1 (ISO-latin1), which correspond to multiple languages. The translation server will attempt to disambiguate these cases by looking first for a language META tag within the document itself. Where no such tag exists, the server will attempt to utilize CRL's statistical language guesser to select the most likely language given the codeset.

The second problem facing the Translation Server is the need to present the document to the Translation Engine in one of the supported codesets. If the document's codeset is not one of those supported by the particular language's translation engine, then the document will be converted by CRL's codeset conversion tool as appropriate. CRL's conversion tool currently supports conversion between approximately 90 of the most frequently used codesets.

Finally, the Translation Server must preserve, as closely as possible, the original HTML formatting of the document during the translation process. An HTML parser supporting HTML version 3.2 is used to extract the textual content of the document for translation. The only formatting that might be lost or misaligned would be non-structural typefacing tags due to non word-for-word translation. For instance, the original document might have a single word highlighted in a group of words translated as a phrasal unit. Since it is often difficult if not impossible to associate a specific source word with its equivalent target word, this type of formatting would be lost.

Conclusion

The emergence of efficient and low-cost MT technology combined with advances in MT architecture (interlinguas, reversibility) make low-cost multilingual MT for assimilation purposes attractive, and the rapid emergence of multinational Web sites and accompanying translation needs seems to be an excellent opportunity to deploy such technology. This is not to say that all problems have been solved, but there are already MT systems using mature elements of this technology. Since all elements of this technology have proven feasible, we believe it now possible to design a MT architecture for assimilation purposes that can support rapid development of multilingual MT systems. The Corelli project[4] is aimed exactly at this.

Acknowledgments. The authors wish to thank Sergei Nirenburg, Jim Cowie, Bill Ogden and Michelle Vanni for their help and support, and for their contribution to the Temple project. We would like to acknowledge the collaboration of Ahmed Malki, Vanishree Mahesh, Nick Ourusoff, Heather Pfeiffer, Susumu Duke Yasuda, Yin Wanying, Daniel Wood, and also many other students who have contributed to this project. Finally, we would also like to thanks anonymous reviewers for their helpful and constructive comments. The Temple project has been funded by the US DoD under grant 94-R-3075/A0001.

References

Callan, J.P.; Croft, W.B.; and Harding, S.M. 1992. The INQUERY Retrieval System. In Proceedings of the 3rd International Conference on Database and Expert Systems Applications, 78-83. Springer-Verlag.

Carrasco Benitez, M.T. 1996. "WInter: Web Internationalization & Multilinguism". INTERNET-DRAFT <draft-benitez-winter-cultures-00.txt>, May 16th, 1996.

Church, K. and Hovy, E. 1993. Good Applications for Crummy Machine Translation. *Machine Translation* 8(4): 239-258.

4. http://crl.nmsu.edu/corelli.

Bryan, M. 1996. Linking HTML Translations. Version 2.0, The SGML Centre.

Davis, M.; Dunning, T.; and Ogden, B. 1995. String Matching Strategies and N-Gram Comparisons. In Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, 27-31. Dublin, Ireland: University College Dublin, Belfield.

Frederking, R.; Grannes, D.; Cousseau, P.; and Nirenburg, S. 1993. An MAT Tool and Its Effectiveness. In Proceedings of the DARPA Human Language Technology Workshop. Princeton, NJ.

Farwell, D.; Helmreich, S.; Jin, W.; Casper, M.; Hargrave, J.; Molina-Salgado, H.; and Weng, F. 1994. Panglyzer: Spanish Language Analysis System. In Proceedings of the Conference of the Association of Machine Translation in the Americas, 56-64. Columbia, MD.

Grishman, R. ed. 1995. Tipster Phase II Architecture Design Document, Version 1.52. New-York University. (http://cs.nyu.edu/tipster)

Kay, M. 1980. The Proper Place of Men and Machines in Machine Translation. Xerox PARC Technical Report, CSL-80-11.

1988. The Penman Primer, User Guide, and Reference Manual. Unpublished USC/ISI documentation.

Matsumoto, Y.; Sadao, K.; Takeji, U.; Yutaka, M.; and Makoto, N. 1993. Japanese Morphological Analysis System, JUMAN. Kyoto University, Nara Science and Technology Graduate School University (in Japanese).

Nicol, G.; Yergeau, F.; Adams, G.; and Duerst, M. 1996. Internationalization of the Hypertext Markup Language. Internet Draft <draft-ietf-html-i18n-05.txt>, Network Working Group.

Nirenburg, S.; Shell, P.; Cohen, A.; Cousseau, P.; Grammes, D.; and McNeilly, C. 1993. Multi-purpose Development and Operations Environments for Natural Language Applications. In Proceedings of the 3rd Conference on Applied Natural Language Processing, page nos. Trento, Italy.

Stein, G.C.; Lin, F.; Bruce, R.; Weng, F.; and Guthrie, L. 1993. The Development of an Application Independent Lexicon: LexBase. CRL Technical Report, MCCS-92-247.

The Unicode Consortium. 1996. *The Unicode Standard*, Version 2.0. Reading, Mass: Addison-Wesley.

Zajac, R. and Vanni, M. 1996. Glossary-Based MT Engines in a Multilingual Translator's Workstation for Information Processing. *Machine Translation*, Special Issue on New Tools for Human Translators. Forthcoming.