WebDIPLOMAT: A Web-Based Interactive Machine Translation System

Christopher Hogan and Robert Frederking
Language Technologies Institute
Pittsburgh, Pennsylvania, USA
chogan@e-lingo.com, ref@cs.cmu.edu

Abstract

We have implemented an interactive, Web-based, chat-style machine translation system, supporting speech recognition and synthesis, local- or third-party correction of speech recognition and machine translation output, and online learning. The underlying client-server architecture, implemented in Java™, provides remote, distributed computation for the translation and speech subsystems. We further describe our Web-based user interfaces, which can easily produce different useful configurations.

1 Introduction

The World Wide Web (Berners-Lee, 1989) seems to be an ideal environment for machine translation: it is easily accessible around the world using freely-available, easy-to-use tools which are available to persons speaking a myriad of languages, all of whom would like to be able to communicate with one another without language barriers. It is therefore not too surprising that a few companies have attempted to make machine translation available in this medium (AltaVista, 1999; FreeTranslation, 1999; InterTran, 1999). The primary use identified for these translators has been that of translating Web pages or amusing oneself with the inadequacies of machine translation (Yang and Lange, 1998). What these systems cannot be used for is real-time, speech-to-speech communication with translation.

Real-time communication over the Internet has more properly been the domain of "chat" protocols: primarily Internet Relay Chat (IRC) (Oikarinen and Reed, 1993), and similar instant messaging protocols developed commercially (America Online Inc., 2000; Microsoft Corp., 2000; ICQ Inc., 1999). While some portals have been developed to permit access to chat using the Web (iTRiBE Inc., 1996), the primary point of access seems to be chat-specific client software. Although chat defines protocols and provides infrastructure, it is limited in the kind of data that it can transport, and client software is tightly focussed on the text domain. Such limitations have not, however, prevented researchers from experimenting with the possibilities of incorporating machine translation or speech into the chat experience (Lenzo, 1998; Seligman et al., 1998). The outcome of these experiments has been to show that commercial machine translation systems may be reasonably integrated into the chat room, and that commercial speech software can be connected to existing chat software to provide the desired experience.

We have taken a different road. It has been noted (Seligman, 1997; Frederking et al., 2000) that broad-coverage machine translation and speech recognition cannot now be useful unless users can interact with the system to improve results. While Seligman et al. (1998) were able to effect user editing of speech recognition by editing text before submitting it for translation, they were unable to do the same for the translation system, primarily due to limitations of commercial software. Additional limitations are encountered in the communication medium: chat is not amenable to non-text interaction with translation agents, and commercial chat software does not, in any case, support such interaction.

To deal with these limitations, we have developed a fully interactive, Web-based, chat-style translation system, supporting speech recognition and synthesis, local- or third-party correction of speech recognition and machine translation, and online learning, which can be used with nothing more than a Web browser and some simple add-ons. All intensive processing, including translation and speech recognition is performed at central servers, permitting access for those with limited computational resources. In addition, the modular design of the system and interface permit computational tasks to be easily distributed and different dialog configurations to be explored.

2 Interface Design

The design of the WebDIPLOMAT system is intended to facilitate the following kind of interaction: (numbers correspond to Figure 1)

1. Speech from the user is recognized and displayed in an editing window, where it may be edited by respeaking or using the keyboard.
2. When text is acceptable to the user, it is submitted for translation and transfer to the other

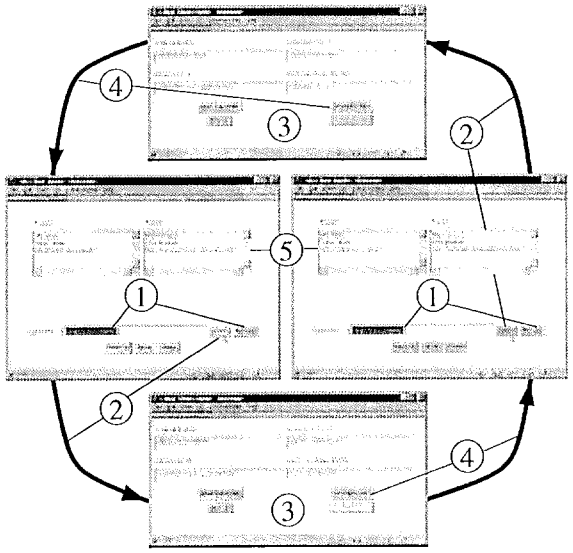

Figure 1: User-level perspective on information flow. See text for explanation of labels.

party.

3. Text to be translated is optionally presented to a human expert, who is able to translate, correct and teach the system a correct translation.
4. Upon machine translation of the text, or acceptance by the expert, a translation is delivered to the other party and synthesized.
5. Both sides of the conversation are tracked automatically for all users, and displayed on their interfaces.

Although the above is the original vision for the system, other configurations are easily imagined. Configurations with more than two participants, or where one of the users is also simultaneously an expert are straightforwardly handled. Internationalization of the interfaces, for use in different locales, is also easily handled. Many changes of this nature are handled by easy modifications to the HTML code for given Web pages. More complicated tasks may be accomplished by modifications of underlying code.

In order to produce the above configuration, the current system implements two user interfaces (UIs): the Client UI, which provides speech and text input capabilities to the primary end-users of the system; and the Editor UI, which provides translation editing capabilities to a human translation expert. In the rest of this section, we describe in detail certain unique aspects of each interface.

2.1 Client User Interface

In addition to speech-input and editing capabilities, the Client UI is able to track the entire dialog as it progresses. Because the Central Communications Server (*cf.* §3.1) forwards every message to all connected clients, every component of the system can be aware of how the dialog turn is proceeding. In the Client UI, this capability is used to provide a running transcript of the conversation as it occurs. By noting the identifiers on messages (*cf.* §3.4), the UI can assign appropriate labels to each of the following: our original utterance, translation of our utterance, other person's utterance, translation of their utterance. In addition, we use knowledge about the status of the dialog to prevent the user from sending several utterances before the other party has responded.

2.2 Editor User Interface

The Editor UI provides tools which make it possible for a human expert to edit translations produced by the machine translator before they are sent to the users. As mentioned earlier, the editing step is optional, and is intended to improve the quality of translations. The Editor UI may be configured so that either of the two users, or a remote third party can act as editor. Our motivations for providing an editing capability are twofold:

- Although our MT system (*cf.* §3.2) does not always produce the correct answer, the correct answer is usually available among the possibilities it considers.
- The MT system provides for online updates of its knowledge base which allows for translations to improve over time.

In order to take advantage of these capabilities, we have designed two editing tools, the chart editor and always-active learning, that enable a human expert to rapidly produce an accurate translation and to store that translation in the MT knowledge base for future use.

As discussed in §3.2, our MT system may produce more than one translation for each part of the input, from which it attempts to select the best translation. The entire set of translations is available to the Web-DIPLOMAT system, and is used in the chart editor. By double-clicking on words in the translation, the

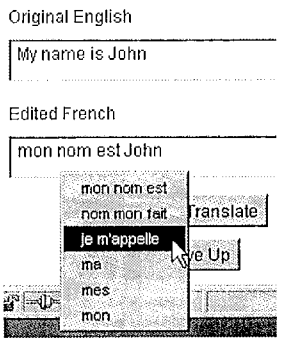

Figure 2: Popup Chart Editor

human editor is presented a popup-menu of alternative translations beginning at a particular location in the sentence (see Figure 2). When one of the alternatives is selected, it replaces the original word or words. In this way, a sentence may be rapidly edited to an acceptable state.

In order to reduce development time, our MT system can be used in a rapid-deployment style: after a minimal knowledge base is constructed, the system is put into use with a human expert supervising, so that domain-relevant data may be elicited quickly. In order to support this, all utterances are considered for learning. When the editor presses the 'Accept/Learn' button, the original utterance and its translation are examined to determine if they are suitable for learning. Currently all utterances for which the forward translation has been edited are submitted for learning, although other criteria may also be entertained. More detail about online learning may be found in §3.2.

Although the editor UI is primarily intended for use by a translation expert, it will sometimes also be used by those who are not as expert. For this situation, we have introduced a backtranslation capability which retranslates the edited forward translation into the language of the input. Although imperfect, backtranslation can often give the user an idea of whether the forward translation was substantially correct.

3 System Design

In this section, we describe the computational architecture underlying the WebDIPLOMAT system.

3.1 Architecture

The underlying architecture of the WebDIPLOMAT system is shown in Figure 3. The system is organized around three servers:

The **Web Server** serves HTML pages to clients. We used an unmodified version of the Apache HTTP Server (Apache Software Foundation, 1999).

The **Speech Recognizer(s)** perform speech recognition for clients.

The **Central Communications Server** allows communication between clients. Encapsulated objects sent to this server are forwarded to all connected clients. With the exception of speech and HTTP, all communications between clients use this server.

The servers are designed to be small, and are intended to coexist on one machine.[1] Currently, however, the speech server includes a full speech recognizer, and therefore consumes a greater amount of resources than the other servers.

Most processing is intended to be performed by clients, which have no locality requirements, and may therefore be distributed across machines and networks as necessary. The User and Editor Clients were described in §§2.1 and 2.2. We will now examine the most important processing mechanisms, including machine translation and speech recognition/synthesis.

3.2 Machine Translation

For Machine Translation, we rely on the Panlite Multi-Engine Machine Translation (MEMT) Server (Frederking and Brown, 1996). This system, which is outlined in Figure 4, makes use of several translation engines at once, combining their output with a statistical language model (Brown and Frederking, 1995). Each translation engine makes use of a different translation technology, and produces multiple, possibly overlapping, translations for every part of the input that it can translate. All of the translations produced by the various engines are placed in a chart data structure (Kay, 1967; Winograd, 1983), indexed by their position in the input utterance. A statistical language model is used, together with scores provided by the translation engines, to determine the optimal path through the set of translated segments, which information is also stored in the chart. Upon completion of translation, the chart data structure is made available for use by the rest of the WebDIPLOMAT system.

Currently, we employ Lexical Transfer and Ex-

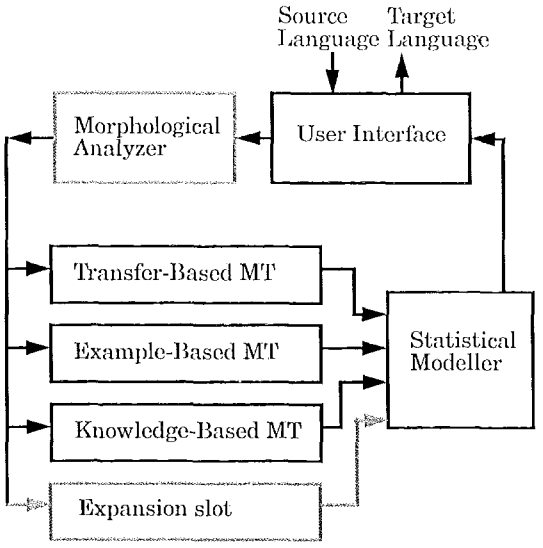

Figure 4: Multi-Engine Machine Translation Architecture

[1] This is necessary due to security restrictions on Java™ Applets.

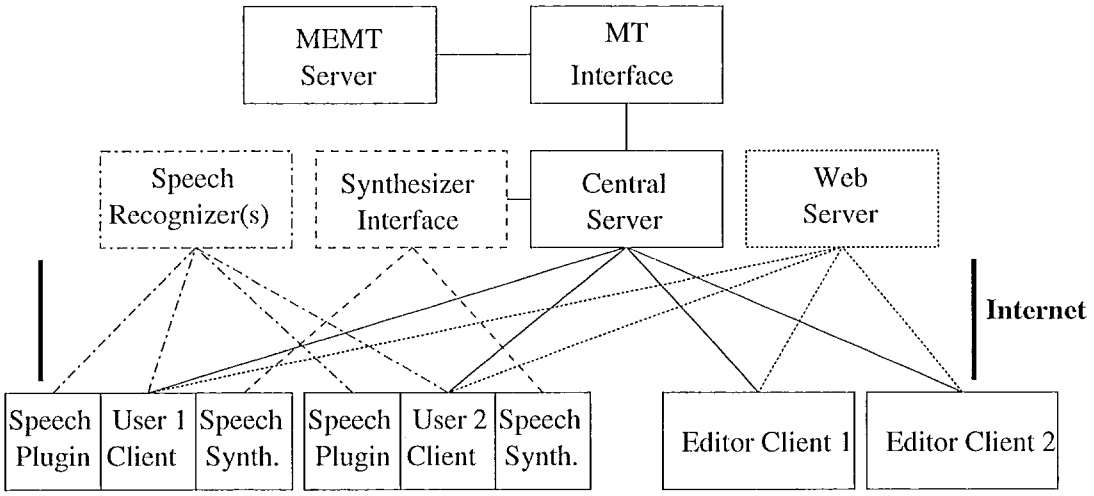

Figure 3: Server Architecture

ample Based Machine Translation (EBMT) engines (Nagao, 1984; Brown, 1996). Lexical Transfer uses bilingual dictionaries and phrasal glossaries to provide phrase-for-phrase translations, while EBMT uses a fuzzy matching step to produce translations from a bilingual corpus of matched sentence pairs. Because the knowledge bases for these techniques are simple, they both support online augmentation. As mentioned in §2.2, the Editor UI attempts to learn from utterances that have been edited. Pairs of utterances submitted for learning to the translator are placed in a Lexical Transfer glossary if less than six words long, and in an EBMT corpus if two words or longer. Higher scores are given to these newly created resources, so that they are preferred.

The MT server is interfaced to the Central Server through MT interface clients, which handle, *inter alia*, character set conversions, support for learning and conversion of MT output into an internal object representation usable by other clients. It also ensures that outgoing translations are stamped with correct identifiers (*cf.* §3.4), relative to the incoming text, to ensure that translations are directed to the appropriate clients.

3.3 Speech Recognition and Synthesis

In the current system, speech recognition is handled as a private communication between a browser plug-in, running on the user's machine, and a speech recognition server, and is not routed through the central server. Speech is streamed over the network to the server, which performs the recognition, and returns the results as a text string. This configuration permits most of the computational resources to be offloaded from the client machine onto powerful remote servers. The speech may be streamed over the network as-is, or it may be lightly preprocessed into a feature stream for use over lower-bandwidth connections. The recognized text is returned directly to the user client for editing and validation by the user before being sent for translation. Our speech server is a previously implemented design (Issar, 1997) based on the Sphinx II speech recognizer (Huang et al., 1992). As mentioned earlier, the speech server and recognizer are not currently designed to run in a distributed fashion.

Unlike speech recognition, which is handled by the User Client, speech synthesis does not require human interaction, and can therefore be connected directly to the central server. Currently, Synthesizer Interfaces unpackage internal representations and send utterances to be synthesized on a speech synthesizer running locally on the user's machine. Future plans call for speech to be synthesized at a central location and transported across the network in standard audio formats.

3.4 Implementation

All components of the WebDIPLOMAT except the speech components and Web Server were implemented in Java™ (Gosling et al., 1996), including the Central Server. Messages between clients are implemented as a Java class `Capsule`, containing a `String` identifier and any number of data `Objects`. Object serialization permits simple implementation of message streams. User Interface clients are developed as Applets, which are embedded in HTML pages served by the Web Server.

4 Future Work and Conclusion

The most significant change we would like to make to the current system is the way that speech is handled. We firmly believe that the best speech input device is the one people are already familiar with, namely the telephone. A revised system would allow users to call specific phone numbers (connected to the central server) in order to access the system, which would then recognize and synthesize speech

over the telephone line while still using web-based interfaces. This, of course, takes us closer to the grand AI Challenge of the translating telephone (OAIAE, 1996; Kurzweil, 1999; Frederking et al., 1999). We contend that by using interactive machine translation, the goal of a broad-domain translating telephone can be more easily brought to fruition.

References

AltaVista. 1999. Babel Fish: A SYSTRAN translation system. http://babelfish.altavista.com/.

America Online Inc. 2000. AOL Instant Messenger(sm). http://www.aol.com/aim/home.html.

The Apache Software Foundation. 1999. The Apache HTTP Server Project. http://www.apache.org.

Tim Berners-Lee. 1989. Information management: A proposal. http://www.w3.org/History/1989/proposal.html, March. CERN.

Ralf Brown and Robert Frederking. 1995. Applying statistical English language modeling to symbolic machine translation. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*, pages 221–239.

Ralf Brown. 1996. Example-based machine translation in the Pangloss system. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*.

Robert Frederking and Ralf Brown. 1996. The Pangloss-Lite machine translation system. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

Robert Frederking, Christopher Hogan, and Alexander Rudnicky. 1999. A new approach to the translating telephone. In *Proceedings of the Machine Translation Summit VII: MT in the Great Translation Era*, Singapore, September.

Robert Frederking, Alexander Rudnicky, Christopher Hogan, and Kevin Lenzo. 2000. Interactive speech translation in the DIPLOMAT project. *MT Journal*. To appear.

FreeTranslation. 1999. FreeTranslation: A Transparent Language translation system. http://www.freetranslation.com/.

James Gosling, Bill Joy, and Guy L. Steele, Jr. 1996. *The Java™ Language Specification*. Addison-Wesley Publishing Co.

Xuedong Huang, Fileno Alleva, Hsiao-Wuen Hon, Mei-Yuh Hwang, and Ronald Rosenfeld. 1992. The SPHINX-II speech recognition system: An overview. Technical Report CMU-CS-92-112, Carnegie Mellon University School of Computer Science.

ICQ Inc. 1999. ICQ IRC Services. http://www.icq.com/.

InterTran. 1999. An InterTran translation system. http://www.airsho.com/transLator3.htm.

Sunil Issar. 1997. A speech interface for forms on WWW. In *Proceedings of the 5th European Conference on Speech Communication and Technology*, September.

iTRiBE Inc. 1996. JiRC. http://virtual.itribe.net/jirc/.

Martin Kay. 1967. Experiments with a powerful parser. In *Proceedings of the 2nd International COLING*, August.

Ray Kurzweil. 1999. *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*. Viking Press.

Kevin Lenzo. 1998. personal communication.

Microsoft Corp. 2000. MSN™ Messenger Service. http://messenger.msn.com/.

M. Nagao. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In A. Elithorn and R. Banerji, editors, *Artificial and Human Intelligence*. NATO Publications.

Office of Artificial Intelligence Analysis and Evaluation OAIAE. 1996. Artificial intelligence--An executive overview. http://www.ai.usma.edu:8080/overview/cover.html.

Jarkko Oikarinen and Darren Reed. 1993. Internet relay chat protocol. ftp://ftp.demon.co.uk/pub/doc/rfc/rfc1738.txt. Request for Comments 1459, Network Working Group.

Mark Seligman, Mary Flanagan, and Sophie Toole. 1998. Dictated input for broad-coverage speech translation. In Clare Voss and Flo Reeder, editors, *Workshop on Embedded MT Systems: Design, Construction, and Evaluation of Systems with an MT Component*, Langhorne, Pennsylvania, October. AMTA.

Mark Seligman. 1997. Six issues in speech translation. In Steven Krauwer et al., editors, *Spoken Language Translation Workshop*, pages 83–89, Madrid, July.

Terry Winograd. 1983. *Language as a Cognitive Process. Volume 1: Syntax*. Addison-Wesley.

Jin Yang and Elke D. Lange. 1998. SYSTRAN on AltaVista: A user study on real-time machine translation on the Internet. In David Farwell et al., editors, *Proceedings of the Third Conference of the Association for Machine Translation in the Americas (AMTA '98)*, pages 275–285, Langhorne, Pennsylvania, October. Springer-Verlag.