NUROP CONGRESS PAPER (A SIMPLE ENGLISH TO CHINESE TRANSLATION WEB SERVICE)

LOW CHIN HONG[1] AND LUA KIM TENG[2]

School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543

ABSTRACT

In view of the rising importance of the Internet and in connected computing, a yet unprecedented Machine Translation system has been built on web service technology in an attempt to make further advances in the research into Machine Translation. This web service capitalizes on the advantages of the new Internet architecture and the flexibility and high customizability and better collaborative use to both users and developers alike.

1 INTRODUCTION

There are plenty of commercial Machine Translation services in use today, many producing translations of a decent quality. But these systems are tightly self-contained, and hardly any room for collaborative development and deployment (Hutchins, 2001). As we step into the next generation of the Internet into connected computing, translation systems will have to be built on new application architectures to fully harness the power of the Internet.

In this project we have built a machine translation system based on web-service technology, new technology built for the next generation of the Internet. The translation system is designed to be a highly modularized, multi-component platform, and is highly customizable, for both end-users and developers alike.

2 NEW TECHNOLOGY, FOR THE NEW INTERNET

The new generation of Internet is powered by this new technology of web services. Web services are self-contained application modules that have open, Internet-oriented interfaces, using HTTP and XML for communications and complex data transfer. Since HTTP and XML are text based, they can be used either behind or outside of firewalls independent of hardware, operating systems, or programming environments – any software client that understands HTTP and XML can make use of its services. This would bring many advantages to research in Machine Translation (MT), as independent developers are now able to bring together their technologies, written in their own language of preference, and make them fit together seamlessly. (Wakefield, 2001)

With these in mind, the plan was then to develop the translation system as a web-service, but not just simply a translator web-service, it would be a translation platform for various translation engines and systems, tools and components to come together, to make translations from the synergies of their strengths. In order to do this, the process of translation has to be

[1] Student

[2] Supervisor

broken down into several phases, so that each of these components can work independently of each other. These components (which could be developed independently) could be plugged in, and the user would be able to choose which combination of components he would like to use to achieve his desired translation.

3 TRANSLATION IN PHASES

In our design and selection of phases, we want to make sure that there are enough phases so that any natural language processing component or even whole systems, past, present or future can fit into one of these phases, but yet small enough such that components do not belong to two phases at the same time.

After analysis of MT systems from the past till the present (SYSTRAN, 2003), we have decided to break down the process of translation into seven distinct phases – input processing, pre-editing, analysis, transfer, post-editing, synthesis and output processing. The lifecycle of translation is briefly covered as follows:

The input documents are filtered and converted into simple text in the *input processing* phase, and is syntactically converted (morphological conversion), or enhanced (tagging) in the *pre-editing* phase to aid in translation in later phases. Information about the text (semantic domain information etc.) is extracted in the *analysis* stage, and might be useful in the *transfer* phase, when the source language text is converted to the destination text. Many different transfer engines can run together in parallel. Each of these outputs is in turn *post-edited* to arrange them in correct lexical order, or simply to clean them for synthesis. In the *synthesis* phase these text are aligned, and synthesize together to get an optimal output text. O*utput-processing* then takes care of returning the text to the user in a format useful to the end-user (Word document, HTML page etc.)

Each of these phases can have any number of components working, and each component works independently of others, each taking in an input and generating one from it. We have designed the system to run all components in sequence, with the exception being the transfer phase, when the engines are fed a common input and run in parallel. Because of its highly modular design, users and developers can mix and match components to use in the translation process, and for components in the same phase, they can decide which order they want the components to be executed.

Although these components work independently of each other, they can reuse code from any other modules as long as they are web-services. As such, even whole systems could be plugged into one phase, and use the tools prior or after to do better machine translation. Basically, any form of natural language processing tool could be included in the system.

4 PROJECT IMPLEMENTATION

We have successfully implemented the translation system using Visual Basic .NET because of better code reusability and better integration with other web-service and languages. For purposes of this project, we have decided to design and implement at least one module per phase (with some having two so we can see the effects of sequencing and parallel execution).

The following have been implemented:

Input processor:

HTML Filter

Pre-Editing:

Spelling Checker (based on large corpus sampling, Levenshtein distance matching)

Word Stemming Engine (based on Porter Stemming algorithm) (Porter, 1980)

Analysis:

Semantic Domain Recognizer (based on Bayes' Probability Theorems) (Jurafsky, 2000)

Transfer:

Direct Translation Engine (based on word group matching)

Post-Editing:

Punctuation Replacement Component

Synthesis

Synthesis Engine (based on CLUSTALW multiple alignment algorithm, and Needleman-Wunsch-Sellers pair-wise alignment algorithm)

Output processing:

HTML Document Generator

A corpus-based translation engine, and a Part of Speech tagger based on HMM algorithms were attempted but not successfully completed. These would belong to transfer phase, and pre-editing phases respectively. Many other modules (information extractors, meaning analyzers, sentence re-constructors) could be implemented if time permits.

As we can see, most of these components represent problems that are still undergoing active research in the realm of natural language process. Any breakthroughs in any of these areas will have a positive impact on our system. And any natural language processing engine can most probably find a place in this system.

5 CONCLUSION AND PROJECT EVALUATION

The components have been tested, and those based on established algorithms were tested for completeness and correctness. The system is capable of simple translations now because of the simplistic transfer engine.

However, we have developed a good platform for collective collaboration in natural language processing and in machine translation, and we have successfully developed a system that works upon this technology. In this project alone, tools and components based on new concepts and ideas, and other based on some well-established concepts have been developed, and shown that even though these components differ in generations, they can work well together.

We feel that this translation system could be used for future developments in machine translation and natural language processing to tested or implemented, to bring research in machine translation and natural language processing to a new level, through collaborative efforts and connected computing.

6 REFERENCE

[1] D. Arnold, L. Balkan, S. Meijer, R. L. Humphreys, L. Sadler (1994), "Machine Translation: an Introductory Guide"

[2] W. J. Hutchins (2001), "Towards a New Vision for MT", Introductory Speech at the MT Summit VIII Conference at Compostela, Galicia, Spain

[3] D. Jurafsky, J. H. Martin (2000), "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" Prentice Hall

[5] M. Porter (1980) "An Algorithm for Suffix Stripping", *Program, Vol. 14*, no. 3, pp 130-137

[6] P. H. Sellers (1974), "On the Theory and Computation of Evolutionary Distances."

SIAM J. Appl. Math., pp. 787-793.

[7] SYSTRAN Software (2003), "SYSTRAN Information and Translation Technologies"

[8] C. Wakefield, H. E. Sonder, W. M. Lee (2001) "VB .NET Developer's Guide", Syngress Publishing, Inc.