French/English Translation

by Sharon Ulery
TJHSST Computer Systems Lab 2006-2007

Abstract

This project uses hard-coding techniques in the field of computational linguistics to translate French to English and English to French well enough to be understandable to someone who knows only the output language.

Introduction

Starting with a word-for-word translation from French to English and vice versa, I gave the program grammar rules so that it correctly translates increasingly complex grammar structures. This project was easy to scale as needed throughout the year. The translation mechanism uses only a small number of grammar rules because encoding extensive rules amounts to data entry. Using the same technique, however, the program could simply be 'taught' to translate all grammatically correct, non-idiomatic sentences in both languages with correct agreement of number and gender, assuming a perfect knowledge of each language's grammar.

Purpose

This project uses computational linguistics to serve students of French or English as a non-native language as well as those who know only one of these languages. The output should be understandable even to someone who knows only the output language. Even a less than perfect translation is useful for surfing the web, reading texts in a foreign language, and communication with someone from another country. It can also be used for students to check their writing by translating back to their native tongue. They can check mechanics and make sure the writing is comprehensible by checking these areas of the translation.

Results

There are two methods of language processing that can be used in translation. Statistical analysis is used in most web-based and commercially available translation programs. Using that method, the translation program "learns" grammar rules by using a few basic rules and going over a large corpus of written material in both languages. This can be contrasted with the method I used, in which grammar rules are hard-coded into a program. The biggest obstacle in using hard-coded grammar rules is that to create a sufficient number of rules and a large dictionary requires a lot of data entry. As this wasn't beneficial in a senior tech lab project, the translation program I wrote is very primitive. However, by expanding the program I created in a straightforward way, further grammar rules can simply be coded in, and the small dictionary I used can be easily expanded or replaced. The program currently fits its specifications exactly, although those specifications are not extensive. If the program were extended through additional data entry, however, it would continue to exactly fulfill specifications as they grew.

Procedures and Methodology

Input: The user works through a GUI to first specify the original language of the text. Then, still through a GUI, the user inputs the phrase to be translated.
Output: Output will be in the terminal. For input using only words in the dictionary and acceptable forms thereof, output is the literal equivalent of the root words in the order they appear in the non-input language.
Requirements: The program is written in Java. It uses a small bilingual dictionary, but a large bilingual dictionary would greatly extend its capabilities A technique for inputing and outputting accent marks would also be valuable.
Presentation of results is relatively simple. A list of a variety of sample inputs and corresponding outputs should allow clear evaluation of the results.
Testing is also relatively simple. As I increased capability, I used specific structural and functional testing of the new grammatical structures the program includes. From time to time, including at the end of the project, I used dynamic testing to make sure that there were no hidden bugs.

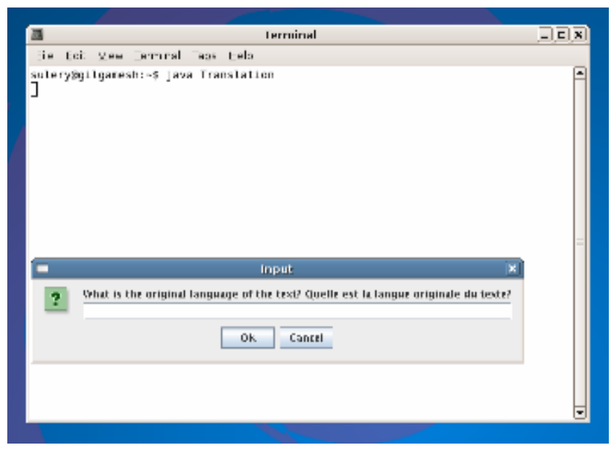
Background

Most current in this area is far above the introductory level of this research project. I read *Foundations of Statistical Natural Language Processing: Chapter 1* by Manning and Schutze. Manning and Schutze believe that a division of language into "grammatical" and "ungrammatical" statements is an artificial and unsuccessful way of looking at it; rather grammatical structures should be seen as more or less commonly used. They use a method such that the program learns the parts of speech of words and common syntactical structures by training it on a large body of input text from a wide variety of fields, because this approach is much more robust than hardwiring all knowledge into the program at the beginning. I also read “Words & Transducers: Morphology, Tokenization, Spelling” and “Machine Translation” from *Speech and Language Processing, 2nd Ed.* 3 by Daniel Jurafsky and James M. Martin. The former source is a good introduction to some of the main problems in artificially "understanding" natural language. It talks about parsing words into morphemes so that a dictionary can be a reasonable size and begins talking about understanding words in context. The latter discusses the problems involved in machine translation and gives an overview of how to address many of these issues. I read Kevin Knight's "Automating Knowledge Acquisition for Machine Translation". This source is an overview of word-for-word statistics-based translation, syntax-based translation and semantics-based translation. It compared the usefulness of each and discussed high-level implementation techniques. Unfortunately, using the full statistical techniques recommended by these sources is far beyond the scope of a year-long, high school project.

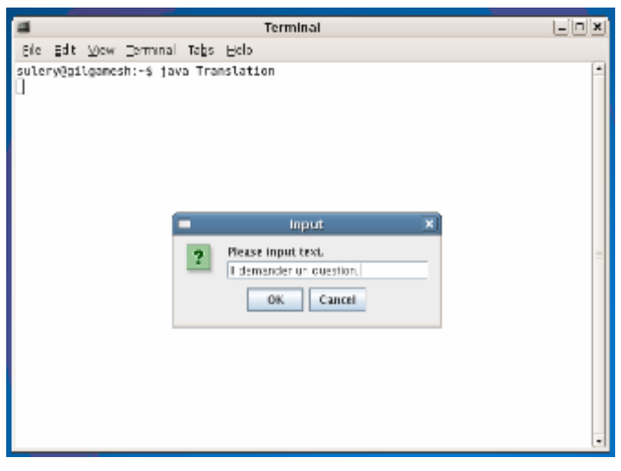
Screen Shots:

Three stages of the Program

1)



2)



3)