Creation of a Russian-English Translation Program

Karen Shiells

April 20, 2006

Abstract

This project will take an object-oriented approach to machine translation. Using dependency trees, rather than the conventional phrase-structure representation, the translator will identify sentence structures and use holes in the parse tree to identify unknown words. Also, rather than making potentially incorrect guesses, the program will ask the user for input, with the goal of aiding a translator fluent in both languages, rather than users unable to read Russian.

1 Introduction

1.1 Purpose

The primary goal of the project is to perform passable machine translation. The translation algorithm will be broken into a lexical scanner to interpret the source text, a syntactic transfer algorithm for the actual translation, and a text constructor for formation of the

output.

The main distinction between this project and commonly available translation algorithms is the intended audience. While machine translation available on the internet is generally designed for individual users who have no knowledge of the source language, and low standards for the translation. This project will produce a program intended for users who are familiar with both languages, and can provide useful input during translation in order to avoid major errors in the output. To this end, the translator will be able to identify unknown words and their structural properties, and present this information to the user to verify and input a literal translation for.

Since the program will ask for aide from the user in interpreting the source language, the finished product will only be useful to those who know the source language, at a minimum, and ideally are familiar with the target language as well. Professional translators, whether working independently or with companies, would benefit from a program that checks its interpretations rather than providing a quick, but low-quality translation. Language students, also, would be able to use this resource while still thinking about the texts, rather than plugging text to be translated into an automatic translator.

1.2 Scope of Study

The program, since it will request user input, will not necessarily need to identify antecedents, grammar structures, and other information as consistently or correctly as programs that attempt to translate independently. Also, since the project also seeks to identify attributes

of unknown words, time spent on the translation algorithm itself should be minimal until all parts of the project have at least been attempted. Many teams of programmers with far more human and capital resources have spent much longer than the time allotted for this project attempting to produce effective machine translation. This project does not attempt to surpass them; only to experiment with translation algorithms and add part of speech recognition to a more primitive translator.

Additionally, the range of grammatical structures, dictionary entries, and particularly idiomatic expressions understood and used by the program will be limited. At the end, if time remains, a few may be added for purposes of experimentation. For the most part, however, the program will parse and produce only simple sentences, representing the most literal translations from the source language to the target.

2 Background

2.1 Machine Translation

Almost all of the research performed in either machine translation or the broader field of computational linguistics could be incorporated into a project similar to this one. Much research has been done on part-of-speech analysis, and far more has been conducted in various methods of machine translation. This project will limit itself to translation by syntactic transfer and word identification using the syntactic structures stored in its dictionary.

2.1.1 Direct Translation

The earliest machine translation algorithms relied on a direct translation approach. They were, essentially, specialized systems, translating directly from one language to another based on a literal dictionary and common structural changes between the two languages, such as shifting from subject-verb-object to subject-object-verb sentence order. Because such algorithms do not so much as parse the original sentence, the output is frequently nonsensical, generally difficult for even humans to parse, and unable to accommodate complex structures. Though sometimes acceptable translators for similar languages, such as French and English, direct translation becomes increasingly incomprehensible as it is expanded to additional languages.

2.1.2 Interlingua

From the age of international languages come the set of interlingua translators. The main advantage of interlingua approaches is the increased flexibility provided by the intermediate representation. Interlingua chooses a representation, often a constructed language or a grammatical diagram, to use as an intermediary. They first translate the source into the intermediary, using algorithms specific to the source language, and then from the intermediary to the target, again using a specialized algorithm. This approach, however, still relies on specific programs for each language, and adds an additional set of translation errors, introduced by the extra language. The wider the range of languages, the more general the intermediary needs to be. As multilingual systems grow, accurate translation becomes harder

to accomplish because forms common to all supported languages are more difficult to find.

2.1.3 Syntactic Transfer

The syntactic transfer method is a more specific complement of interlingua. Syntactic transfer first uses a language-specific parser to generate a structural tree from the source text. The structural tree is then rearranged to remove any grammatical peculiarities of the original language. Using the resulting tree, a second program generates a text in the target language. Like interlingua, syntactic transfer provides an intermediary representation of the text, though it is specific to the particular language pair. Though syntactic transfer can produce the best quality translations, however, it is relatively new, and not as practical for programs with many language pairs. Syntactic transfer has not been the subject of as much experimentation as the other methods, partially because of legacy code, but also because it is less practical for more profitable multilingual translators.

Because only one language pair will be translated, this project will use a primarily transfer-based approach. Whereas other research projects in the Computer Systems laboratory have been largely based on direct translation, this project will attempt to place an emphasis on parsing the input, rather than expanding the dictionary immediately. A syntactic transfer approach can, by generalization, be expanded into an interlingua approach. This project will attempt to remain flexible enough that it may later be adapted to use interlingua.

2.2 Syntactic Structures

2.2.1 Valency

As an aide for parsing, and for identification of unknown words, the program will use a valency dictionary that is, a dictionary that provides the number and types of complements that each verb takes.

2.2.2 Phrase-structure Representation

The syntactic structures most people are familiar with are phrase-structure representations. Phrase-structure representation breaks sentences into noun phrase, verb phrase, and other superstructures, and from there into the component words.

2.2.3 Dependency Structures

To aide valency analysis, the parser will use a dependency grammar, which produces a tree in which nouns are dependent on verbs, adjectives are dependent on nouns, and so on, using words as nodes, rather than the superstructures in phrase-structure representation.

3 Procedures

3.1 Approach

Because only one language pair will be translated, this project will use a primarily transfer-based approach. Whereas other research projects in the Computer Systems laboratory have

been largely based on direct translation, this project will attempt to place an emphasis on parsing the input, rather than expanding the dictionary immediately. A syntactic transfer approach can, by generalization, be expanded into an interlingua approach. This project will attempt to remain flexible enough that it may later be adapted to use interlingua.

3.2 Components

The first program necessary for the project to proceed will be the dictionary, hopefully including a simple Java program to help review and generate entries. Since the parser uses the dictionary so heavily, including its details, the dictionary will need to be fully programmed in order to proceed with the translator itself. After the dictionary, the lexical analysis algorithm can be written, probably still in Java. The syntactic analyzer will be in either Python or Java, and will generate and store the dependency tree that will form the intermediary representation. Finally, the target language generator, back in Java, will create the English text from the tree. After these are all operating on a basic level, the preliminary version of the unknown word analyzer can be added to the syntactic analysis stage.

3.3 Expansion

In the second stage of development, attributes, including a separate program to manage attribute relations, can be added. Initially, part of speech and possibly gender will be the only attributes included. Additional attributes, such as verb of motion, animate or inanimate, and other specifiers can make translation more accurate and more likely to select

the correct interpretation of each sentence. Both the lexical and syntactic analysis stages, as well as word identification, would need to be updated to include attributes. If the project reaches this point, the dictionary and parsing algorithms can be improved independently for as long as time allows.

4 Results

4.1 Word Identification

Because Russian has fairly complex morphology, one section of the program deals exclusively with identifying the root forms of words. For regular nouns and verbs, this process is relatively easy. However, as spelling rules, irregularities, and other variations are added to the system, word identification gets complex and time-consuming. Though this program continued to use reverse-morphology to find a word before checking for it, performing some sort of analysis to find a stem in common, then checking the conjugated forms would probably be more efficient.

4.2 Parsing

The dependency parse tree works fairly well, though the particular class structure in this project makes it somewhat messy. Some issues arise with implied subjects and verbs, but since the structure is more flexible, sentences with reflexive verbs, multiple verbs, and other oddities are easier to parse. The method of storing all structures, however, takes up a lot of

space, and could certainly be streamlined.

4.3 English Generation

English generation presents somewhat of a problem, since the English information must either be stored in the dictionary along with the Russian, glossed over and left to poor grammar, or hard-coded in the program. None of these are particularly attractive, and all three add significantly to the space requirements of the program, but the dual storage is most acceptable from a software engineering perspective, so this program took that route. As the program turned out, an alternate target-generation method could be substituted for the English one, but the dictionary would need to be updated and probably additional grammatical issues would arise.

5 Conclusions

The program works fairly well at what it is intended to do. Its abilities, however, are fairly limited, and cannot expand beyond the code. More recent natural language processing programs usually used statistical approaches, with which I am not yet familiar. Since these do not need perfect information, and presumably can identify words either from roots or from less information in the code and the text, they would be more able to expand once coded.

5.1 Recommendations

To improve the program, the first and simplest change would be to incorporate statistical language processing. Of course, even without doing that, many grammatical and semantic rules could be added. The program ignores punctuation, and does not consistently identify which noun or verb prepositional phrases are modifying. Semantic information, also, would help with this, though I have used none in my program.

References

[1] Allen, James. Natural Language Understanding. New York: Benjamin/Cummings Publishing Company, 1995.

[2] Arnold, Doug, Lorna Balkan, Siety Meijer, R. Lee Humphreys, and Louisa Sandler. Machine Translation: An Introductory Guide. London: NCC Blackwell, 1994. Available Online: http://www.essex.ac.uk/linguistics/clmt/MTbook/PostScript.

[3] Barber, Charles. The English Language: A Historical Introduction. Cambridge: Cambridge University Press, 1993.

[4] Beard, Robert. "Russian: An Interactive On-Line Reference Grammar". November 1, 2005. Available Online: http://www.alphadictionary.com/rusgrammar/.

[5] Comrie, Bernard, ed. The World's Major Languages. Oxford: Oxford University Press, 1990.

[6] Hutchins, John and Harold Somers. An Introduction to Machine Translation London: Academic Press, 1992. Available Online: http://ourworld.compuserve.com/homepages/WJHutchins/IntroMT-TOC.htm.