TJHSST Computer Systems Lab Senior Research Project
Development of a German-English Translator
2007-2008

Felix Zhang

January 23, 2008

Abstract

Machine language translation as it stands today relies primarily on rule-based methods, which use a direct dictionary translation and at best attempts to rearrange the words in a sentence to follow the translation language's grammar rules to allow for better parsing on the part of the user. This project seeks to implement a rule-based translation from German to English, for users who are only fluent in one of the languages. For more flexibility, the program will implement limited statistical techniques to determine part of speech and morphological information.

Keywords: computational linguistics, machine translation

1 Introduction - Elaboration on the problem statement, purpose, and project scope

A perfect machine translation of one language to another has never been achieved, because not all language expressions used by humans are grammatically perfect. It is also infeasible experimentally to code in every single grammar rule of a language. However, even a basic program that translates the basic idea of a sentence is helpful for understanding a text in a given language.

1.1 Scope of Study

I will focus on a rule-based translation system, because of time and resource constraints. I will start with part of speech tagging and lemmatization, and then progress to coding in actual grammar rules so that sentences can be parsed correctly, so that my program can handle more complex sentences as I embed more rules. I will also expand the program to incorporate limited statistical methods, including part of speech tagging and linguistic property tagging. At best, the program should be able to translate virtually any grammatically correct sentence, and find some way to resolve ambiguities.

1.2 Purpose

The goal of my project is to use rule-based methods input to provide a translation from German in to English, or vice versa, for users who only speak

one of the languages. Though the translation may be simple, the program still aids a user in that it provides a grammatically correct translation, which facilitates understanding of even primitive translations. Basic translations of short passages are especially helpful for users reading less formal text, as sentence structures tend to be less complex.

2 Background and review of current literature and research

Rule-based translation is the oldest form of language processing. A bilingual dictionary is required for word-for-word lookup, and grammar rules for both the original and target language must be hard coded in to structure the output sentence and create a grammatical translation. Most online translators currently are based off of SYSTRAN, a commercial rule-based translation system.

The more modern technique, statistical machine translation, is the most-studied branch of computational linguistics, but also the hardest to implement. Statistical methods require a parallel bilingual corpus, which the program reads to "learn" the language, determining the probability that a word translates to something in a certain context using Bayes Theorem:

$$\tilde{e} = arg \max_{e \in e^*} p(e|f) = arg \max_{e \in e^*} p(f|e)p(e)$$

They can also be used to determine linguistic properties, such as part-of-speech and tense. Usually, statistical methods are more accurate when the corpus used is larger (Germann, 2001). Statistical methods are considerably more flexible than rule-based translation, because they are essentially language-independent. Google Translate, which has access to several terabytes of text data for training, currently is developing beta versions of Arabic and Chinese translators based on statistical methods. Most research is being done with much more funding and resources than my project, and is thus much more advanced than my scope.

3 Development

The main components to a rule-based translator are a bilingual dictionary, a part of speech tagger, a morphological analyzer that can identify linguistic

properties of words, a lemmatizer to break a word down to its root, an inflection tool, and a parse tree.

3.1 Dictionary

The dictionary stores a German word, its part of speech, its English translation, and any other data relevant to its part of speech, for example, for nouns, it also lists its plural form and gender. A large dictionary would be impractical for testing purposes, so I only include pronoun forms, conjunctions, and articles, with only a few nouns and verbs. These entries are stored in a hashtable, with German words as keys and English translations as values.

3.2 Part of speech tagging

The program first attempts to tag words in the input sentence using the freely available TIGER corpus, which consists of 700,000 German tokens, with each token manually assigned a part of speech. For large, full sentences, the program stores the entire corpus into a hashtable. Each unique word in the corpus serves as a key, while each table value is a list of tuples. Each tuple represents a different part of speech assigned to the word in the corpus. The first element in the tuple is the part of speech, while the second is a number, indicating the frequency of the tag's occurrence. For single words and short phrases, it is more efficient to search for the single word in the corpus, and incrementing a separate counter for the occurrences of each different part of speech assigned to it. When a word, usually a noun or verb, is unable to be looked up in the corpus, a rule-based system is used as backoff. These rules are specific to the language being translated. For example, if a word is in between an article and a noun, it will be tagged as an adjective.

3.3 Morphological Analysis

Morphological analysis would use definite articles, suffixes, and adjective endings to determine linguistic properties such as gender, case, tense and person. It generates possible pairs of gender and case for nouns, and tense and conjugation for verbs. Two separate sets of pairs are generated for articles and modifiers, and the final list of possibilities is derived from the intersection of these two sets. To reduce ambiguity, a method for noun-verb

agreement is used to determine the subject of the sentence. This information is used for lemmatization.

Morphological analysis can also be implemented statistically. Since each token in the TIGER Corpus is also assigned linguistic information such as gender, case, and number, the likelihood of a word having certain linguistic properties can be calculated. The simplest calculation would be for gender, since singular words will not change gender in different contexts.

3.4 Noun-verb agreement

Since each word will often generate several different possibilities during morphological analysis, a method for noun-verb agreement is used. The properties of the nouns nearest to the verb in the sentence are crosschecked with the properties of the verb, according to conjugation. A singular noun, if next to a singular third-person verb, will most likely be the subject of the sentence. This method helps to disambiguate verbs and nouns, by reducing the possibilities of gender, case, tense, and person.

3.5 Lemmatizer

The lemmatizer takes information from the morphological analysis and breaks a word down into its root form. For nouns, this means that plural nouns should be reduced to singular form, and suffixes resulting from different grammatical cases should be removed. When the program encounters a word that may be plural, it attempts to remove any of the common verb endings from the word: -e, -en, -er, -ern, and -s. For verbs, any ending from conjugation or tense should be removed. The program takes the few possible conjugation endings, "-e", "-st", "-t", and "-en", removes them, and adds "-en" to the root to render the infinitive form of the word. The prefix for past-tense verbs, "ge-", is also searched for and removed. This saves considerable space in the dictionary, as I do not have to code in every inflected form of every word.

3.6 Parse tree

The parse tree arranges the sentence based on dependency grammar. Verbs connect from the subject to the direct object, and articles and adjectives are nodes of nouns. In translation, this tree must be rearranged to accommodate the target language's grammar.

3.7 Inflection

Since the dictionary lookup will only produce the root form of the translated word, a simple inflection tool is used to conjugate words, once translated into English. Inflection requires the information from the morphological analysis, which it then uses to add endings to words. Words marked as plural add an "-s" or "-es" to the end, as do singular verbs, depending on whether the root word ends in a consonant or a vowel. Also taken into account are common ending changes, such as words ending in "-y" turning into "-ies" in the plural.

4 Testing

Testing is conducted through input of sentences with new features. To test my lemmatizing component, I would input various inflected forms of a word to check the uniformity of the program's output. To test part of speech tagging, two versions of a corpus are needed, one tagged and one untagged. The program attempts to tag all words in the untagged corpus, which is then checked against the manually tagged corpus for accuracy. Varying sentence structures can also serve as a functional test to check the validity of newly coded grammar rules in the parse tree.

5 Results

My program is able to translate a simple German or English sentence into the other language, provided the word is known in the lexicon. A statistical tagger correctly resolves most ambiguities in words. The project fulfills its purpose as a simple translator with basic grammar rules and basic statistical techniques, but would need an implementation of more advanced statistical methods to attain more flexibility in sentence structure parsing.

5.1 Word ambiguity

In German, many words can be taken to very different meanings depending on the contexts. For example, the German pronoun "sie" can be translated to "she", "her", "they", "them", or "you". Though the program does attempt to resolve as many ambiguities as possible using noun-verb agreement, there still exist cases wherein even a native human speaker of German would have

trouble disambiguating, such as a sentence in which both nouns could possibly be the subject.

5.2 Encoding Problems

A characteristic unique to the German language is the use of special characters in its alphabet, such as diacritic marks. Due to program constraints, these characters can not be expressed directly during input, instead substituting them for their closest equivalents: ö is expressed as "oe", while ß is expressed as "ss". An issue with the corpus lay in the corpus compilers' attempt to encode the special characters, which ended up as garbled ASCII code when the corpus was read into the program.

5.3 Corpus Size

Though a larger corpus typically allows for greater accuracy in tagging, file size can be a constraint in many cases. The TIGER Corpus, consisting of 700,000 lines, is 42 megabytes in size, making it impractical for web-based or portable use. The amount of time spent by the program while going through the corpus also presents a problem of convenience and efficiency.

5.4 Stem changes

In general, most inflected verbs in German add a suffix, depending on its conjugation - first person singular adds an "-e", second person singular adds an "-st", and third person singular adds "-t". However, for several exceptions in German, the root word itself alters slightly in singular conjugations. For example, the verb "lesen", which means "to read", has a vowel change when conjugated in the third person singular, "er liest", as opposed to the expected "er lest". Only certain verbs follow this rule, which means the program cannot simply change the vowel stem when it encounters such a conjugation, but the verbs that express this quality are too commonly encountered to simply disregard. A way around this problem is to include an indicator in the dictionary entry for the word, noting that the verb is irregularly conjugated.

Similarly, German verbs are divided into "strong" verbs and "weak" verbs. Weak verbs follow a common pattern in the present perfect tense, adding a "ge-" prefix and a "-t" suffix. The program's morphological analysis easily detects weak verbs. Strong verbs, however, have no set pattern

when in the past tense, including many vowel changes. For strong verbs, the only way to resolve the problem is by manually including the past tense form for each strong verb in the dictionary.

5.5 Statistical accuracy

According to Charniak (1997), when assigning part-of-speech statistically, the accuracy of tagging should approach 90 percent when each word is simply assigned its most frequently occurring tag. Running the part-of-speech tagger on the sample corpus confirms this, yielding accuracy of around 90 percent.

References

[1] Brants, Thorsten, "TnT: a Statistical Part-of-Speech Tagger", *Applied Natural Language Conferences*, pp. 224-231, 2000

[2] Charniak, E, "Statistical Techniques for Natural Language Parsing", *The American Association for Artificial Intelligence*, pp. 33-43, 1997

[3] Germann, U, "Building a Statistical Machine Translation System from Scratch", *Proceedings of the Workshop on Data-driven Methods in Machine Translation*, pp. 1-8, 2001

[4] Tiger Corpus, http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus/

[5] Lezius, W, "A Freely Available Morphological Analyzer, Disambiguator and Context Sensitive Lemmatizer for German", *Proceedings of the COLING-ACL*, pp. 743-748, 1998