TJHSST Computer Systems Lab Senior Research Project
Statistical Machine Translation (Spanish to English)
2009-2010

Raghav Bashyal

April 9, 2010

Abstract

Statistical Machine Translation (SMT) aims to learn a language much the same way a human would naturally by comparing a translation to its original text and attempting to associate words between the two. This project aims to build such a program. Although SMT implementations usually are capable of translating to and from any language, this study will focus on Spanish and English. Using Natural Language Toolkit and Python, this project aims to create a translator as described by Kevin Knight. It would then adjust the programming as well as the input to test the effectiveness of new and existing techniques. It would also test the effectiveness of combining syntax based translation, which is translation facilitated by hard-coded rules, with SMT.

Keywords: Statistical Machine Translation, Spanish, English

1 Introduction

1.1 Scope of Study

The purpose of this project is to successfully implement Statistical Machine Translation techniques to translate from Spanish to English and to test the effectiveness of new techniques such as hard-coded syntax.

1.2 Expected results

This project should be able to translate text from Spanish to English accurately, and also able to learn continuously form input data. The analysis and effectiveness can be presented by displaying sample translating with highlighted errors and with simple charts that show the frequency of such errors. The program should be able to identify some of its own errors in translation by using a reference-only database. Adjustments in the program, such as hard-coded components of the translation process or an algorithm meant to simply a procedure will be tested to see if they yield better translation results.

1.3 Type of Research

My research is in part applied research, as it focuses a lot on the implementation of the translation system. It does have components of use-inspired research as I will be testing various techniques to improve the translations made, which could be used for real-life research.

2 Background and Review of Literature

This project requires me to become familiar with the Natural Language Toolkit, a free tool commonly used for projects involving Natural Language Processing. to become familiar with the field and with the tool, I have read Statistical Machine Translation by Adam Lopez and Getting Started with Natural Language Processing with Python by Nitin Madnani. These pieces gave me an idea of what areas the field incorporated. Christina Wallin implemented the tool last year in her research paper: Naive Bayes Classication, where she tested a a new technique for classifying news into categories. In Improving English-Spanish translation, Preslav Nakov implemented techniques

like paraphrasing and expanding recasing and tokenization with improved translations, although smaller text turned out worse than larger text. I am also working with the NLTK book to gather the knowledge required to implement my ideas.

Using Kevin Knight's guide "A Statistical MT Tutorial Workbook," I am learning the process and implementing at the same time. The Statistical Machine Translation process begins with the calculation of probabilities for words and phrases. These probabilities form the foundation for more complex conditional and Bayes probabilities that will be used to determine the likelihood of certain words that may occur next to one-another, which is part of the N-gram model. This model can apply smoothing coefficients that will implement the machine-learning part of the program. With the smoothing, the program has polished the preliminary probability-calculation stage. Using perplexity and logarithmic adjustments, which modify the probabilities so that they are on a normal plane of comparison, this section could be further improved.

2.1 Perplexity

This process is used to minimize the effect of a large data file on a small probability. To prevent them from become incomprehensibly small, perplexity can be used to normalize the values. "As P(e) increases, perplexity decreases. A good model will have a relatively large P(e) and a relatively small perplexity. The lower the perplexity, the better." (Knight) This will be useful later, when unique models will be compared with each other.

2.2 Log Probability Arithmetic

The Log Probability Arithmetic is a way of preventing the P(e) from underflowing (due to the numbers being so small). Using converted logarithmic values saves the numbers in manageable sizes, keeping the values from underflowing.

3 Procedures and Methodology

The Natural Language Toolkit and its auxiliary packages will compose this project. In addition to the functions that it provides, it has a system that

allows mass amounts of data - texts, in this case - to be input in blocks called corpuses. I will be using the provided tools to translate and the corpuses for testing. The testing is fairly simple since the only thing that needs to be done is to compare the results to the available translations or checked manually for accuracy.

So far, the program accepts text input either from the NLTK database, which can be accessed using import, and with self-made corpuses, for which NLTK has special functions. It can take the input form the corpus and tokenize it by sentences so that the program can later work with clean strings. The program is also able to calculate the probability of words present in a certain amount of text, which will build up to the larger probabilities needed later. Although this is done pretty accurately, the range of the program is limited. It then uses smoothing to adjust the probabilities calculated so that those that are calculated using singular words and bigrams are held at a lower level of significance than trigrams, which have three words to produce a confident probability. Other pieces of the program include the Guided User Interface, which is in its preliminary stages and will be used for the going through singular steps of the translation process.

Currently my procedure is to study the NLTK book and practice with the problems in it. I will also be doing the same thing with the SMT guidebook by Knight. After I reach a point from which I can jump off into my real work, I will write physical pieces of my program.

4 Expected Results

The results from this project would be shown in highlighted errors in translated text, as well as charts that plot the frequency of errors. The results would also show whether or not the variations implemented in the techniques provided positive results. A display of the probabilities of contending models would also be provided by the GUI.

This project could be improved by implementing algorithms not implemented and changing things so that they may produce better results.

5 Bibliography

Charniak et. Al. "Syntax-based Language Models for Statistical Machine Translation." Department of Computer Science, Brown University; Information Sciences Institute, University of Southern California.

Knight, Kevin. "A Statistical MT Tutorial Workbook." April 1999. JHU Summer Workshops.

Fonollosa, J. and Khalilov, Maxim. "N-gram-based Statistical Machine Translation versus Syntax Augmented Machine Translation: comparison and system combination." Proceedings of the 12th Conference of the European Chapter of the ACL, pages 424-432, Athens, Greece, 30 March - 3 April 2009. c 2009 Association for Computational Linguistic.

Melamed, Dan. "Algorithms for Syntax-Aware Statistical Machine Translation." Computer Science Department. New York University.

Palmer, Martha and Wu, Zhibiao. "Verb Semantics and Lexical Selection." National University of Singapore; University of Pennsylvania