Spanish-English Machine Translation

Ian Fisher and Henry Mohr

Our project was an effort to develop a basic machine translation tool from Spanish to English using some of the algorithms that we studied in class. Our tool runs in four stages: part-of-speech tagging, parsing, syntactic transfer, and lexical transfer. The part-of-speech tagger uses the TBL algorithm, where the tagger is trained on a pre-tagged corpus to learn a set of transformations that are then applied to every new text. The parser implements the Earley algorithm, with a handwritten grammar. We experimented with extracting the grammar automatically from the treebank, but the resulting grammar was far too large to be useful in practice. For syntactic transfer, we just implemented a simple Noun Adjective → Adjective Noun transformation as a proof of concept. Lexical transfer uses a dictionary lookup, with some special code to handle verb conjugations.

We evaluated our project manually, for the most part, as machine translation is an especially difficult problem to verify automatically. The tagger was tested against the Spanish corpus with which it was trained and got 97.8% of the words correct. However, this number should be taken with a grain of salt as the tagger is bound to do better with the training corpus than with any other. If this project were solely to implement the tagger, we would have tested it more rigorously, but since the tagging algorithm was only a small part of the entire architecture, we decided that a less accurate but still indicative test would suffice.

The syntactic parser is definitely the area where the translator fails the most. For better accuracy, we could try to pare down our automatically-extracted grammar and improve the parser's speed to acceptable levels. We would also want to implement more syntactic transformations to better reflect the differing grammar of English and Spanish.

Our project is entirely encapsulated in a single function: `translate` in the `translate.py` module. The function takes a sentence as a string, and returns a list of possible translations. This module should be run with Python 2 and `nltk`. Also, if you want to play around with the taggers, you can import `cess_tagger` and `brown_tagger` from `tbl.py` and use the methods `cess_tagger.tag_word` and `cess_tagger.tag`. The grammar can be loaded by importing `CFGrammar` from `cfg.py` and doing `CFGrammar.from_file('grammar.txt')`.