IMPROVING MACHINE TRANSLATION WITH SENTIMENT PRESERVATION IN HINDI-BANGLA PAIR

Rajarshi Roychoudhury

February 29,2019

Synopsis:

Apertium provides an engine and toolbox that allows users to build their own machine translation systems. It uses rule-based machine translation.

The goal of my project is to improve machine translation by preserving the sentiment of a sentence. It has been seen in many cases that the original sentence tries to convey a certain sentiment , while the translated sentence conveys an entirely different sentiment , and gives an incorrect translation (usually seen in the cases of sarcastic sentences).For example:the sentence “Its funny how thieves try to break into a house and get arrested .” when translated to Esperanto in Apertium gives “Ties amuza kiel ŝtelistoj provas rompi en domo kaj akiras arestita ”, which means the thieves try to both break into a house and get arrested. But the original sentence refers to a completely different meaning.

The idea of my project is to use sentiment analysis to analyse the pattern of such sentences and form a rule to give better translation of these sentences.For this i propose to introduce a new <sdef> symbol called “sentiment” in the monolingual dictionary. While introducing a

word in the monolingual dictionary I will also store the sentiment of that word(discussed later). The <pattern> tag will analyse the sequence of such sentiments , and will decide which rule to implement. For example, say we have 3 classes of sentiment:positive(1) , negative(0) and neutral(2). The input sentence when encoded on the basis of sentiment is “1101112” (say). The goal is to utilise this pattern to make rules for better translation (say maintaining the same pattern , or rearranging it in translation to incorporate the sentiment).

For sentiment analysis I will take the help of neural nets.Since Apertium deals with low resource languages , a huge corpus for sentiment analysis is not available for word level embedding. But for any language the number of unique characters is well deterministic. Since words are just sequences of characters , I will determine weight vector for character level embedding , use that sequence of character embeddings to form an embedding for a word, and feed it into a Recurrent Neural Network for classifying the sentiments of words. I have conducted experiments on this , and received an accuracy of 63% for a corpus of 8000 words. This method **can be used for any language** . For obvious reasons , this classification task will be done in Python , and the result will be written in a text file. I will then read the file using C++ , so in entirety the result will be independent of Python.

The rest is same as developing a new language pair.I want to work on **two related languages -Hindi and Bangla(which share the same Sanskrit root) .** Previously some work has been done in Hindi-Bangla translation in apertium , though not much. I want to improve that as well as incorporate the above mentioned ideas.