SanTran

Ben Walczak, Brandon Kupczyk, Charlie Morley, Shivam Thakrar, and Shivani Kohli

Web-page: https://bkupz.github.io/SanTran-WebPage/

Project Description

The aim of SanTran is to train an English-Sanskrit sentence translation model. Translation is needed in a variety of settings: businesses, cultural media, governments, and social platforms can all benefit from efficient and effective automated translation, in addition to individuals. Sanskrit is one language still missing effective tools for full translation. There are numerous online dictionaries for Sanskrit-English word-by-word translation, but there is no full-fledged translator which takes the context and grammar of sentences into account. According to the 2001 census of India, Sanskrit has about five million speakers (as a first, second, or third language), so an effective, automated written translator of full sentences would be widely useful.

Machine Translation is an active but well-developed field of data science; there are already a host of techniques and models that have been tried and tested on numerous languages for full translation. Applying some of these to Sanskrit specifically would be challenging and interesting. Sanskrit has a non-Latin alphabet, but also an intermediate “Transliteration” language - called IAST - which is made up of Latin characters. This could help lower the complexity of the task.

Project Goals

The goal is to create a machine learning model that can translate to and from a few different languages. As we begin the project, we plan to first do simpler translations such as English-German. This will allow our model to learn the semantics of different languages and be able to, therefore, translate more complex languages. Thus, we will be able to adapt what we have developed to English-Sanskrit translations. Sanskrit translations will be our final goal because little to no progress has been done with the translation of it compared to other languages.

By starting with more common languages, we can compare our model with many other well-accepted models and tools to evaluate if our model is performing well, before moving onto Sanskrit. In addition, we plan to use the typical strategy of splitting the data into training and testing subsets to determine if our model performs well.

The model we will be using is commonly known as Encoder-Decoder LSTM, where LSTM refers to Long-Short Term Memory. LSTM is a specialized version of a recurrent neural network. It works by understanding each word in a sentence based on your understanding of previous words. The context of the sentence helps aid the understanding of what each word within the sentence should be. Recurrent neural networks use loops to allow information to persist in future occurrences. LSTM allow for learning of long-term and short-term dependencies. The model also consists of an encoder-decoder, two submodels. The encoder is responsible for generalizing and summarizing the semantics between many languages. Whereas the decoder is responsible for predicting an output sequence within the given language, one character per iteration of the recurrent neural network.

Another potential modeling structure that is being considered to do the translation is a Statistical Machine Translation. Statistical Machine Translation uses statistical models based on an analysis of bilingual text corpora. In our case we plan to use morphological information about Sanskrit pulled from the first data source below and other English morphological sources. There are existing tools such as the *Morphogen* which assists the translations between morphologically rich and destitute languages.

Tools

TensorFlow is a widely used Python package for evaluating machine learning models such as neural networks. We plan to use TensorFlow as the primary technology for implementing our model because of its simplicity (it integrates into the popular and easy language, Python), its popularity (many prototypes for machine translation online seem to be written using this tool), and its power (the tool connects with CUDA, a system for hardware acceleration). We may test other deep learning modules within Python if we have sufficient. MXNet, for example, has the ability to utilize multiple processors during training, which may be beneficial considering the complexity of the model and the size of the data.

Data set:

http://sanskrit.inria.fr/manual.html

The Morphological Tagging can be used in Statistical Machine Translation

http://sanskritbible.in/

Full verse-by-verse translation of the New Testament, (has English and Sanskrit side-by-side). We can query the website for Json objects containing the pairs of text. Can be used for Sequence to Sequence LSTM.

http://opus.nlpl.eu/

Has a massive repository of parallel translations between English and German, French, Spanish, and lots of other common languages (a little bit of Sanskrit, but not a lot). Can be used for Sequence to Sequence LSTM.

https://old.datahub.io/dataset/sanskrit-english-lexicon

Dataset created for NLP modeling contains Sanskrit to English translation of words. Can be used for Sequence to Sequence LSTM.

https://doi.org/10.5281/zenodo.803508

Dataset created for Sanskrit Word Segmentation. They developed the dataset to remove the confusions and redundancies present in the general Sanskrit words. For instance, अहं गच्छा (anh: gaccha)and गच्छामि (gacchami) mean the same thing.

https://www.holy-bhagavad-gita.org/chapter/2/verse/1

Bhagavad Gita in Sanskrit translated to English. Contains both the transliteration to IAST and translation to the English language.

Literature

Amrith Krishna, Pavankumar Satuluri, and Pawan Goya, *A Dataset for Sanskrit Word Segmentation*

The authors have released a dataset which contains 80,000 sentences that have been correctly segmented to remove the inconsistencies that otherwise occur in Sanskrit to English translation as Sanskrit is a language in which words are collated and if broken down incorrectly could have opposite meanings. The authors have created the dataset to make Sanskrit more accessible to the computing field as they have undertaken the basic preprocessing that we would otherwise have to undertake.

Rashmi Jha, Aman Jha, Deeptanshu Jha, and Sonika Jha, *Is Sanskrit the most suitable language for Natural Language Processing?*

This paper discusses if Sanskrit is a language fit for NLP. The authors conclude that Sanskrit is a viable language for NLP as compared to other languages its grammar and syntax is easier to understand. Additionally, as words like it is, and often I do not have separate words in Sanskrit but can be easily decoded due to inflections it reduces a four-letter English word to 2 letters in Sanskrit making it easier to decode and reducing storage space. Lastly, the authors found that due to the well thought out structure of the language with easy to follow grammatical rules, it makes it possible to correctly translate and decipher the language.

Lee, Young-Suk, *Morphological Analysis for Statistical Machine Translation* (IBM)

This paper discusses the use of an older technique of translation called Statistical Machine Translation but with the addition of Morphological features. These Morphological and syntactic features are also called 'part-of-speech' tags. At its core this paper takes the corpus of each languages and utilizes the part-of-speech tagging to create features for the statistical analyses and model. More specifically this paper utilizes prefix and suffix information and part-of-speech mapping to power their model.

Priscila Aleixo, Thiago Alexandre Salgueiro Pardo, *Finding Related Sentences in Multiple Documents for Multidocument Discourse Parsing of Brazilian Portuguese Texts*

The goal of this paper was to find some measure that allows for similar sentences to be found in multiple documents that are the same topic. After pre-processing the sentences such that all the sentences have their stopwords removed and the whole document is lemmatized, a lexical similarity measure was applied. After all the different measures were taken, the authors looked at recall, precision, f-measure, and cosine similarity all along with various thresholds to see which measure provided the best results (of which recall was preferred). The measures included tests such as "word overlap + stoplist" and "cosine + lemmatization". There were two main conclusions that can be drawn from the study. The first being that the best measure that provided the best results, under the assumption that recall is the better measure, was cosine similarity + lemmatization which had about a 93-100% recall around a threshold of 0.1-0.2. The second conclusion was that the results were overall better when looking at documents in Portuguese as opposed to English for Brazilian Portuguese texts.

P Bahadur, A Jain, D.S.Chauhan, *English to Sanskrit Machine Translation*

This paper describes the main differences between English and Sanskrit, and then goes onto explain a general algorithm of translating between the two. It goes into great detail of how we can use a top-down processing style, where one can start with the Sanskrit text -> break it up into tokens and phrases -> parse them -> take that and parse it into English grammar -> and finally convert that phrase into English. Finally, the paper discusses how the various parts of speech in the English language (noun, verb, adjective, adverb) is stored into a database with added tokens to the word to make recognition easier for the machine.

References:

Amrith Krishna, Pavankumar Satuluri, and Pawan Goya, *A Dataset for Sanskrit Word Segmentation*

Lee, Young-Suk, *Morphological Analysis for Statistical Machine Translation* (IBM)

P Bahadur, A Jain, D.S.Chauhan, *English to Sanskrit Machine Translation*

Priscila Aleixo, Thiago Alexandre Salgueiro Pardo, *Finding Related Sentences in Multiple Documents for Multidocument Discourse Parsing of Brazilian Portuguese Texts*

Rashmi Jha, Aman Jha, Deeptanshu Jha, and Sonika Jha, *Is Sanskrit the most suitable language for Natural Language Processing?*