Sentiment Analysis on Code-mixed data from Twitter

Mayank Goel, 2019114004
Tejasvi Chebrolu, 2019114005
Team Cookie Monster
Advanced NLP Project

Abstract:

In this project, we perform the challenging task of sentiment analysis on Codemixed Tweets. After implementing a Naive Bayes model as our baseline, we show results that are comparable to similar efforts in this field, using Multilingual BERT and XLM-RoBERTa. We also implement approaches using Machine Translation, using both a Transformer-based model (with dismal results due to lack of a sizable corpora) and a novel approach using dictionary-based translation.

Introduction:

At the sentence or utterance level, code-mixing refers to the employment of linguistic components such as words, phrases, and clauses from various languages. It is most commonly witnessed in a casual atmosphere, such as on social media. The amount of code-mixed data available to us is enormous, thanks to the numerous social media platforms available for individuals to communicate. Sentiment analysis (or opinion mining) is a natural language processing (NLP) technique used to determine whether data is positive, negative or neutral. While most efforts in NLP in general are focused on English data, in this project we worked with code-mixed data. This made the task substantially more challenging.

Architecture:

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. In its vanilla form, Transformer includes two separate mechanisms — an encoder that reads the text input and a decoder that produces a prediction for the task. As opposed to directional models, which read the text input

sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it's non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word).
Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. In technical terms, the prediction of the output words requires:

- Adding a classification layer on top of the encoder output.
- Multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension.
- Calculating the probability of each word in the vocabulary with softmax.

XLM-R uses self-supervised training techniques to perform cross-lingual understanding. It improves upon previous multilingual approaches by incorporating more training data and languages — including so-called low-resource languages, which lack extensive labeled and unlabeled datasets. The model uses the same shared vocabulary for all the languages. This helps in establishing a common embedding space for tokens from all languages. Hence, it is evident that languages that have the same script (alphabets), or similar words map better to this common embedding space.

For tokenizing the corpora, a Byte-Pair Encoding (BPE) is used. a Transformer architecture with 1024 hidden units, 8 heads, GELU activations (Hendrycks and Gimpel, 2016), a dropout rate of 0.1 and learned positional embeddings. We train our models with the Adam optimizer (Kingma and Ba, 2014), a linear warmup (Vaswani et al., 2017) and learning rates varying from 10^{-4} to 5.10^{-4} were the other parameters.

Methodology:

The task is to predict the sentiment of a given code-mixed tweet. The sentiment labels are positive, negative, or neutral, and the code-mixed languages will be English-Hindi. The data is arranged in the ConLL format, and we had to do pre-processing to both make it into a suitable format for our model, as well as to clean the tweets to improve our model accuracy.
Our baseline was a Naive Bayes classifier for multinomial distributions. As expected, our model didn't perform too well and had an accuracy of ~43%.
We then used pre-trained multilingual embeddings from BERT. For instance, BERT embeddings have been shown to perform very well on multiple downstream tasks. BERT, or Bidirectional Encoder Representations from Transformers, improves on ordinary Transformers by removing the unidirectionality requirement by pre-training with a masked language model (MLM). The masked language model masks some tokens from the input at

random, with the goal of predicting the masked word's original vocabulary id based solely on its context. The MLM aim, unlike left-to-right language model pre-training, permits the representation to integrate the left and right context, allowing us to pre-train a deep bidirectional Transformer.
After this, we implemented a XLMR model, and a binary XLMR model, which is the current state of the art model for multilingual tasks. We implemented the binary model because we realised that the neutral sentences were what was bringing the F1 score down.
We had another approach planned around Machine Translation, with the intuitive idea that sentiment analysis on English text would be easier than codemixed text. However, the parallel corpora available was of insufficient size, and we had to abandon the approach as the results weren't good. We used a Transformer architecture for the same.

Slides And More

https://docs.google.com/presentation/d/1NtAWvLOp7_momZI3wokIeoBBQl2bOXd1KDvTrbbnXgE/edit?usp=sharing

References

- https://competitions.codalab.org/competitions/20654
- https://arxiv.org/abs/2008.04277
- https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- https://www.researchgate.net/profile/Vinodhini-G-2/publication/265163299_Sentiment_Analysis_and_Opinion_Mining_A_Survey/links/54018f330cf2bba34c1af133/Sentiment-Analysis-and-Opinion-Mining-A-Survey.pdf
- https://arxiv.org/pdf/1901.07291.pdf
- https://arxiv.org/pdf/1804.00806.pdf