

Query Labelling for Indic Languages using a hybrid approach

Rupal Bhargava¹ Yashvardhan Sharma² Shubham Sharma³
Abhinav Baid⁴

Department of Computer Science & Information Systems
Birla Institute of Technology & Science, Pilani, Pilani Campus
{¹rupal.bhargava, ²yash, ³f2012493, ⁴f2013018}@pilani.bits-pilani.ac.in

ABSTRACT

With a boom in the internet, social media text has been increasing day by day. Much of the user generated content on internet is written in a very informal way. Usually people tend to write text on social media using indigenous script. To understand a script different from ours is a difficult task. Moreover, nowadays queries received by the search engines are large number of transliterated text. Hence providing a common platform to deal with the problem of transliterated text becomes really important. This paper presents our approach to handle labeling of queries as part of the FIRE2015 shared task on Mixed-Script Information Retrieval. Tokens in the query are labeled on basis of a hybrid approach which involves rule based and machine learning techniques. Each annotation has been dealt separately but sequentially.

Keywords

Transliteration, Natural Language Processing, Language Identification, Machine Learning, Logistic Regression, Information Retrieval

1. INTRODUCTION

There are a large number of indigenous scripts in the world that are widely used. By indigenous scripts, we are referring to any language written in a script that is not Roman. Due to technological reasons such as a lack of standard keyboards for non-Roman script, the popularity of the QWERTY keyboard and familiarity with the English language, much of the user generated content on the internet is written in transliterated form. Transliteration is the process of phonetically representing the words of a language in a non-native script. For example, many times to represent a colloquialism such as □□□ (Okay) in Hindi, users will write their transliterated form [1]. Search engines get a large number of transliterated search queries daily – the challenge in processing these queries is the spelling variation of the transliterated form of these search queries. For example the Hindi word □□□□ can be written as ‘khana’, ‘khaana’, ‘khaanna’, and so on. This particular problem involves the following: (1) Taking care of spelling variations due to transliteration and (2) Forward/Backward transliteration. Similarly, with the rise in the use of social media, there has been a corresponding increase in the use of hashtags, emoticons and abbreviations. So, along with identification of languages, these need to be recognized as well. Also, named entities should be considered separately [2].

2. SUBTASK 1: QUERY WORD LABELING

Suppose that $q: w_1 w_2 w_3 \dots w_n$, is a query written in the Roman script. The words, $w_1 w_2$ etc., could be standard English words or ¹<http://www.ark.cs.cmu.edu/TweetNLP/>

transliterated from another language $L = \{\text{Bengali (bn), Gujarati (gu), Hindi (hi), Kannada (kn), Malayalam (ml), Marathi (mr), Tamil (ta), Telugu (te)}\}$. The task is to label the words as en or a member of L depending on whether it is an English word, or a transliterated L-language word. Further Named Entity (NE) recognition and identification of mixed language words (MIX) and Punctuation (X) also had to be carried out.

3. PROPOSED TECHNIQUE

Our system reads the input file and separates them into tokens. After identification of all the tags, an output is generated for the same. We collected more data for Gujarati and Hindi from previous year’s Microsoft FIRE event for the training purposes. Logistic regression was used to train each language individually. Feature set used for the same included unigram and bigram character index with unigram contributing the most in our opinion. Rule based approach was used for combining the individual language classifiers, based on the probability obtained. For other annotations, the process is explained as follows in their respective stages.

The token identification (X, NE, Mix etc.) is done in a pipelined manner. The 4 stages of the pipeline are:

- 1. Identification of Punctuation (X):** The tag X encompasses all forms of punctuation, numerals, emoticons, mentions, hashtags and acronyms. This stage can further be divided into 2 parts done sequentially – identification of emoticons, hashtags, etc. and identification of abbreviations.
 - a. Identification of hashtags, emoticons, etc.:** This is done using the CMU Ark tagger¹ with a training model especially designed for social media text. The tagging model is a first-order maximum entropy Markov model (MEMM), a discriminative sequence model for which training and decoding are extremely efficient [4].
 - b. Identification of abbreviations:** A dictionary based approach is used for this purpose. A list of around 1400 commonly used abbreviations in SMS language was built and the word was marked as X if it occurred in this list.
- 2. Identification of Named Entities (NE):** Named entities were also identified using a dictionary based approach. The training data was used to create the dictionary of Named entities because the data was insufficient to run a machine learning algorithm. The number of named entities was 2414. The number of Named Entities was too low and the multi-language nature of the dataset made it hard to characterize words as NE with certainty.

For example, in English language named entities occur in certain manner at certain positions according to sentence structure. But when it comes to multi lingual sentences, sentence structure varies a lot.

3. **Identification of Language:** For language detection, the classifier was built using Logistic Regression with feature vectors containing character unigrams and bigrams [3].
4. **Identification of mixed words (MIX):** Finally, a rule based approach was adopted for identifying mixed words in the utterances. If the 2 maximum language probabilities in the list generated in the previous stage are close to each other, then the word was classified as MIX. The threshold for detecting MIX words was determined empirically. The threshold was 0.05 with word length greater than 8. It was determined empirically by setting it at different values and manually evaluating the output.

If there is a match in stages 1 or 2 of the pipeline, then the token is immediately abbreviated and no further stages are implemented on that word. Otherwise, the token passes through stages 3 and 4 above so that the final tag can be determined.

4. EXPERIMENTS AND RESULTS

We used the data given to us which included labeled utterances from social media and blogs to build our training data set. We submitted three runs, where we used char 1, 2 - grams as features.

We manually removed a few words from the named entity list in run 2. In run 3, mixed word detection was enabled; it was disabled in the other runs to avoid accuracy from going down to due to false positives. Our training data consisted of 41882 words including all languages and named entities. The training data set was built as a dense model i.e. data is represented using 0 for those features that are not present in the word, and 1 for those that are present, with the feature vector containing 712 entries per word corresponding to each possible character 1-gram and 2-gram. A separate model was built for each language containing an equal number of words in the language and words not in the language. We used the scikit-learn toolkit¹ for machine learning [5]. For language identification, we tried linear regression, naïve Bayes and Logistic Regression classifier.

We used an 80-20 split of the training data to test the performance of our system for cross validation on our test set. The results (shown in table 1) obtained using the evaluation script for our individual classifiers were:

Table 1: Language wise Precision for different classifiers on test data from the 80-20 split

¹ <http://scikit-learn.org/stable/>

	Linear	Naive Bayes	Logistic
en	0.8577	0.7653	0.8660
bn	0.7545	0.7528	0.7605
ta	0.7176	0.7762	0.7642
mr	0.7263	0.7432	0.7402
kn	0.7415	0.7375	0.7298
te	0.7920	0.7542	0.7626
ml	0.7883	0.7622	0.7582
gu	0.6697	0.7501	0.6968
hi	0.7343	0.7138	0.7391
Avg.	0.7536	0.7506	0.7575

The result calculated above were evaluated using the script provided. The results showed clearly that the individual classifiers were pretty good. We decided to use a linear kernel for logistic regression as it was giving the highest accuracy. We tried out different parameters and choose the configuration most optimal for our training data.

Table 2: Official language wise F-Measure, Precision, Recall

Language	F-Measure	Precision	Recall
X	0.8237	0.8963	0.7619
br	0.4803	0.4327	0.5397
en	0.7214	0.6171	0.8683
gu	0.0849	0.1784	0.0557
hi	0.3853	0.3473	0.4326
kn	0.4038	0.4281	0.3821
ml	0.297	0.3896	0.24
mr	0.3141	0.3899	0.263
ta	0.5365	0.6501	0.4567
te	0.3444	0.3473	0.3415

Our overall performance was:

Table 3: Weighted F-Measure and token accuracy for the three runs.

tokens	11999	11999	11999
tokens Correct	6576	6575	6574
Weighted FMeasure	0.567742	0.56769	0.567615851
tokens Accuracy	54.8046	54.7962	54.7879

As shown in Table 3 our overall Weighted F-Measure was 56.7%. Also, our standard deviation was close to 10% error margin.

In addition there was a direct correlation in the results between the precision and the training data sizes used. The number of words for the different languages in the training data was 3509 (bn), 17392 (en), 744 (gu), 4237 (hi), 1520 (kn), 1126 (ml), 1868 (mr), 3116 (ta) and 5960 (te).

As shown in Table 2, Languages like English for which the training data size was larger gave around 72% f-Measure and 87% recall with 61% precision, while Gujarati which had very less training data gave 17% precision. We did better on the weighted F-Measure statistic because the languages with less training data were also the ones least represented in the test data. As such weighted evaluation of the language predictor gave us around 56% F-Measure.

Named Entity recognition was done based on a lookup based method that would classify words as named entities in the test set if they were found in the training set. This was done because the training set for named entities was too small to use a machine - learned Named Entity Recognizer. The results obtained by our approach reaffirmed that our approach was correct.

It was observed that the Language Predictor developed based on our approach inaccurately predicted on testing data due to the small training data. The precisions of our individual classifiers and the official results for English, Bengali, and Tamil back our claim.

5. CONCLUSION AND FUTURE WORK

In this paper, we discussed the n-gram approach to identify the language of a word. The context cues of the word could be used to identify the language instead of only relying on character unigrams and bigrams. A future work could be to implement a sequence based classifier that would classify the word based on the previous and the next word. Instead of using only unigrams and bigrams, the system could be improvised to use {1, 2, 3, 4, 5}grams based on different machine learning algorithms such as MaxEnt, Naïve Bayes, Logistic regression, SVM, etc. Our Named Entity recognizer was prone to errors due to insufficient data. Similarly, the accuracy of our system could be improved by training it on more data. However, X tokens were identified with a reasonable accuracy.

Tagging of MIX words could also be improved by using better thresholds.

6. REFERENCES

- [1] King, Ben, and Steven P. Abney. "Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods." HLT-NAACL. 2013.
- [2] Parth Gupta, Kalika Bali, Rafael E. Banchs, Monojit Choudhury, and Paolo Rosso. 2014. Query expansion for mixed-script information retrieval. In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval (SIGIR '14). ACM, New York, NY, USA, 677-686. DOI=<http://dx.doi.org/10.1145/2600428.2609622>
- [3] Spandana Gella, Kalika Bali and Monojit Choudhury. "Ye word kis lang ka hai bhai?" Testing the Limits of Word level Language Identification. (To appear) In Proceedings of the Eleventh International Conference on Natural Language Processing (ICON 2014). Goa, India.

- [4] Owoputi, Olutobi, O'Connor, Brendan, Dyer, Chris, Gimpel, Kevin, Schneider, Nathan and Smith, Noah A. "Improved part-ofspeech tagging for online conversational text with word clusters." Paper presented at the meeting of the Proceedings of NAACLHLT, 2013.
- [5] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.