Cross Language Training and Classification

Nishant Kumar	*Anirban Mishra*	*Sayali Degaonkar*	*Pooja Bhandari*
7305-8548-36	*6287-3467-22*	*3434-3196-94*	*4146-0109-87*
nishant@usc.edu	*anirban@usc.edu*	*degaonka@usc.edu*	*poojab@usc.edu*

1. Introduction

1.1 Rationale

Although there has been a lot of research work done on monolingual text classification in various domains, bi-lingual or cross language text classification (CLTC) remains to be a less explored area of Natural Language Processing. CLTC is defined as the task of training the model in one language and then classifying the test data in other language based on our trained model.

For our research we worked on Hindi and Marathi languages both of which belong to Devanagari script. We manually collected and annotated real world news data (Sec 2.1) in Hindi and Marathi belonging to these four categories: Politics, Sports, Entertainment and Business. We trained our model (Sec 2.2.5) on Hindi news data and used this model to classify the Marathi news articles.

1.2 Challenges

As both the languages (Hindi and Marathi) belong to the same script, we need not had to worry about the challenges arising from transliteration, which usually is the case with languages belonging to different scripts. The biggest challenge in our study is the wide difference in the spelling of the words having same or similar meaning, in the two languages. As shown in Fig 1 below, we can see the difference in the way word "Disappointment" is spelled in Hindi and Marathi. Also Marathi literature uses different forms of the same word. For e.g. "From the policy" (Table 1) can be written as one word in Marathi, whereas it needs two words in Hindi.

Hindi/English Transcription	Marathi/ English Transcription	English
निराशा/Nirasha	नैराश्य/Nairashya	Disappointment
योजना से/Yojna Se	योजनांचे/ Yojananche	From the policy

Table 1

This makes it difficult to define the features used for classification since we can't directly use the word from one language as the feature to be tested on a model trained on the other language.

Some of the other challenges that we faced were unavailability of corpus as needed for research and no relevant bilingual dictionary. Also as there was little or no work done on this topic so it was interesting to see how far we succeed.

1.3 Related Work

Our work closely relates to [1] which uses model translation and Expectation Maximization for CLTC, and [2] which uses bilingual dictionary but all these works are done on English, Chinese and Italian languages. We learnt from and built upon these concepts and have designed and developed our own stemming algorithm (Sec 2.2.7) suited to Hindi and Marathi text classification.

2. Method

2.1 Materials

Since we could not figure out any relevant news corpus in Hindi and Marathi which is classified into the categories that we decided to work on, we collected the data manually by referring to various news websites and archives in Hindi and Marathi. Another big reason for collecting the data manually is the necessity of a "Parallel and Comparable Corpora" in Hindi and Marathi. A "Parallel and Comparable Corpora" means that all the articles collected in a particular language (e.g. Hindi) should be from similar topics and approximately same timeframe as the articles collected in the other language (e.g. Marathi in our case).

If we don't use comparable corpora, the results of classification of news articles on the two different languages will not make much sense as although the categories are same, we are trying to classify two text which refer to totally different topics. Timeframe is also an important factor for collecting news data as the data might be quite biased depending on the period which we are considering. For e.g. if we collect political news data in US newspapers from a period of Jan-2016 to Apr-2016, most of the articles would be referring to US presidential elections. It won't be justified to compare these articles with articles belonging to early 2015.

So we collected a comparable corpora of 2000 news articles with 250 articles each belonging to each category in both the languages (250 * 2[language] * 4[categories]).We also analyzed the volume of data(total size) in each category, since if one category has a very large articles in 250 files, the final result would automatically tend towards that category.

Apart from this we built our own bi-lingual Marathi-Hindi dictionary of around 15000 high frequency Marathi words which was used while classifying the articles in Marathi.

2.2 Procedure

2.2.1 Classification Algorithms: Naïve Bayes and SVM

2.2.2 Features: The features used in both the algorithms are words from the news data. While classifying we either directly use these features or apply the stemming algorithm on the word as discussed in Sec 2.2.7. The reason we can't directly use these features is there are very few words with exactly same spelling in both Hindi and Marathi.

2.2.3 Tools: scikit for SVM classification and calculating F1 score.

2.2.4 Data Preprocessing: We did tokenization, removed punctuations and stop words on both Hindi and Marathi news data to clean it.

2.2.5 Model:
Naïve Bayes: Model is a dictionary with unique words in the training corpus as keys their count as values.
SVM: Every news article is represented as a vector of all unique words obtained from the training data with values being term frequencies. Thus, a news is a data point in n-dimensional space where n is the number of unique words. SVM constructs a set of hyper-planes in this space, to be used for classification.

2.2.6 Annotations: As described in Sec 2.1 we collected and annotated the data manually into Politics, Business, Sports, Entertainment in both the languages.

2.2.7 Stemming Algorithm: Since the words with same meaning in Hindi and Marathi spell differently, but have same roots, we have developed a stemming algorithm where we generate word cluster by truncating the word from the end. Table 2 below shows the word cluster generated for the Marathi word "Chitrapatatil" (meaning of a movie). Here the cluster has 4 words out of which the word "Chitrapat" matches the Hindi model.

Marathi	Latin transcription	Translation
चित्रपट	Chitrapat	Movie
चित्रपटा	Chitrapata	-
चित्रपटात	Chitrapatat	In a movie
चित्रपटाती	Chitrapatati	-

Table 2

2.2.8 Algorithm

- Preprocess both Hindi and Marathi data (Sec 2.2.4).
- Train the model (2.2.5) for both Naïve Bayes and SVM based on annotated corpus in Hindi.
- Search Marathi word in the model for the matching word.
- If word is not found in the model look for the word in Marathi-Hindi dictionary for its Hindi translation.
- If not found in both above two steps, obtain word cluster(Sec 2.2.7) for the word, and then try mapping the words in the cluster with the model.
- If all above cases fail then we do smoothing (add one/ Laplace smoothing) for NB and ignore the word for SVM.
- Calculate the posterior probability for each text file for NB and build news vector for SVM and classify the files.

2.3 Evaluation

2.3.1 Baseline Model We have taken Naïve Bayes with exact Marathi word as feature as our baseline model. With this model we got our F1 score as 0.44.

2.3.2 Measures for evaluating the system's performance

- **Preprocessing:** Removing Stop words and punctuations: F1 score increased to 0.45.
- **Using Dictionaries:** Despite the same script, Hindi and Marathi language contains completely different words for the same meaning. E.g. Book in English is "किताब"(Kitab) in Hindi and "पुस्तक"(Pustak) in Marathi.Some Hindi words are not present in Marathi language and vice versa. Perhaps, some Hindi words have completely different meaning in Marathi language. This made algorithm to do smoothing for most of the words. We overcame this issue by using bilingual dictionary. We used Marathi to Hindi dictionary containing around 15000 words, which increased F1 score to 0.637.We could even achieve higher efficiency by increasing vocabulary size.
- **Using Stemming:** Since many words in Hindi and Marathi have same root word, but word changes the form in both languages. E.g. the root word for court in both languages is "न्यायालय"(Nyayalaya).When the sentence takes the form "by the court", Marathi word takes form "न्यायालयाने"(Nyayalayane) and Hindi takes form "न्यायालय ने"(Nyayalaya Ne). Most of the Hindi words take the root word as it is. Marathi words combines the phrases in single word. That's why we create a word cluster by applying the stemming algorithm (2.2.7). This step increased F1 score to 0.908.

- **Named Entity Recognition:** We ran NER over our whole corpus to make sure that our algorithm is actually learning from the model. When we classified after removing the named entities from the corpus, we got F1 score of 0.856.

3. Results

Previous work done (References) on CLTC suggests that with a reliable bilingual dictionary and a comparable corpora the algorithm is expected to achieve the efficiency of a monolingual classifier.
After applying our algorithm (Sec 2.2.8) described above we got the following count of files classified correctly as in Table 3:

Category	Files Correctly Classified
Business	210/250
Politics	216/250
Entertainment	226/250
Sports	248/250

Table 3

As we can see from above table the sports section was getting categorized most efficiently whereas the business and politics section got some of the classification wrong. The next step in analyzing the result was to find if named entities each category affect the classification of files. For doing this we removed the named entities from each category like names of famous personalities, places, currencies and countries (Table 4).

Business		Entertainment		Politics		Sports	
डॉलर	Dollar	कपूर	Kapoor	भारत	India	गेल	Gayle
दिल्ली	Delhi	संजय	Sanjay	ओबामा	Obama	ऑस्ट्रेलिया	Australia
भारतीय	Indian	खान	Khan	राहुल	Rahul	इंडीज	Indies
मुंबई	Mumbai	सलमान	Salman	मोदी	Modi	चार	four
जेटली	Jaitley			गांधी	Gandhi	कोहली	Kohli

Table 4

After the removal of the above (Table 4) named entities we got the following count of correct classification in each category.

Category	Files Correctly Classified
Business	205/250
Politics	200/250
Entertainment	220/250
Sports	244/250

Table 5

As we can see above (Table 5) the classification is still close to the values that we achieved in Table 3. So this shows that our classifier is able to learn from the algorithm and it is not named entities that lead to the correct classification.

We also took the F1 measure for both classification algorithms used, i.e. the Naive Bayes and the SVM implementation:

NB: [Business-0.89397089, Entertainment-0.93801653, Politics-0.85882353, Sports-0.94095238]
SVM: [Business-0.49707602, Entertainment-0.55555556, Politics-0.56916996, Sports-0.7606679]

The reason for F1 score = 0.90 in Naïve Bayes is Naive Bayes can handle unknown words in model by using bilingual dictionary, word clustering and smoothing.

In case of SVM, we tuned our features by removing stop words, using dictionary and word clustering. We got F1 score of 0.63. After carefully analyzing the model, we realized that the number of features for SVM are very huge in number compared to the training samples (We considered 16821 words as features).This results in a very sparse feature vector for SVM and hence low score.

4. Discussion

4.1 Conclusion

In this research project we tried to do cross language text classification and found the process to be feasible for Hindi and Marathi data. We found the accuracy to be close to any monolingual classifier. Even after removing named entities like name, place, organization and location the accuracy of classification remained close to any monolingual classifier. Just using the Naive Bayes with smoothing resulted in very low accuracy but after the use of a manually created bilingual dictionary and our own word clustering algorithm we were able to match the accuracy reached by any monolingual classifier. The algorithm used by us is simple and effective and seems to be a viable solution for multilingual classification problems that use a common script.
From the research project it can be concluded that with the use of a bilingual dictionary and a word clustering algorithm we can classify related languages that share a common script.

4.2 Contribution to NLP Community

As stated earlier there was no corpus readily available to work on, we built our own corpus and manually annotated the data. So any future work in the same domain in NLP community can use our corpus (https://github.com/nishant4498/CrossLanguageClassifier/). We created our own Marathi-Hindi dictionary which has a collection of 15000 words. This can be used as a bilingual dictionary for further research work.

As a part of our research project we presented a clean and easy to understand algorithm which easily does cross language classification with good accuracy.

4.3 Future Work

As a part of future work we would be working on refining our algorithm to achieve better accuracy. We would be also working on refining the clustering algorithm to find clusters of more accurate words in order to improve efficiency of our algorithm. The other area of focus will be on implementing word sense disambiguation along with our algorithm and we think that it would further improve the efficiency. Further work can be done to check the efficiency of our algorithm on other languages like Gujarati with Bhojpuri which share common script.

References

[1] LeiShi, Rada, Mihalcea and Mingjun Tian. 2010. Cross Language Text Classification by Model Translation and Semi-Supervised Learning. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing.
[2] Alfio Gliozzo and Carlo Strapparava. 2006. Exploiting Comparable Corpora and Bilingual Dictionaries for Cross-Language Text Categorization. Proceedings of the 21st International Confer-ence on Computational Linguistics and 44th Annual Meeting of the ACL.
[3] Manoj Kumar Chinnakotla, Sagar Ranadive, Pushpak Bhattacharyya and Om P. Damani. 2007. Hindi and Marathi to English Cross Language Information Retrieval. At CLEF 2007.

Division of Work

Corpus Annotation: Anirban & Nishant (Hindi), Sayali & Pooja (Marathi)
Background Research: Anirban, Sayali
Algorithm & Feasibility Analysis: Nishant, Pooja
Bilingual-Dictionary formation: Pooja, Sayali
Implementation: Anirban, Sayali, Pooja, Nishant
Documentation: Nishant -25%, Anirban-25%, Sayali- 25%, Pooja- 25%

Word Count 1987