LREC 2018 Workshop

WILDRE4– 4th Workshop on Indian Language Data: Resources and Evaluation

PROCEEDINGS

Edited by

Girish Nath Jha, Kalika Bali, Sobha L, Atul Kr. Ojha

ISBN: 979-10-95546-09-2
EAN: 9791095546092

12 May 2018

Proceedings of the LREC 2018 Workshop
"WILDRE4 – 4th Workshop on Indian Language Data: Resources and Evaluation"

12 May 2018 – Miyazaki, Japan

Edited by Girish Nath Jha, Kalika Bali, Sobha L, Atul Kr. Ojha

http://sanskrit.jnu.ac.in/conf/wildre4/index.jsp

Acknowledgments: This work has received funding from the Microsoft Research India.

Organising Committee

- Girish Nath Jha, Jawaharlal Nehru University, India
- Kalika Bali, Microsoft Research India Lab, Bangalore
- Sobha L, AU-KBC, Anna University

Workshop Manager

- Atul Kr. Ojha, Jawaharlal Nehru University, India

Programme Committee

- Adil Amin Kak, University of Kashmir, India
- Anil Kumar Singh, IIT-BHU, India
- Arul Mozhi, University of Hyderabad, India
- Asif Iqbal, IIT-Patna, India
- Bogdan Babych, University of Leeds, UK
- Claudia Soria, CNR-ILC, Italy
- Dafydd Gibbon, Universität Bielefeld, Germany
- Delyth Prys, Bangor University, UK
- Dipti Mishra Sharma, IIIT, Hyderabad India
- Diwakar Mishra, EZDI, Ahmedabad, India
- Dorothee Beermann, NTN Univeristy(NTNU), Norway
- Elizabeth Sherley, IITM-Kerala, Trivandrum, India
- Esha Banerjee, Google, USA
- Eveline Wandl-Vogt, Austrian Academy of Sciences, Austria
- Georg Rehm, DFKI, Germany
- Girish Nath Jha, Jawaharlal Nehru University, New Delhi, India
- Hans Uszkoreit, DFKI, Berlin, Germany
- Jan Odijk, Utrecht University, The Netherlands
- Jolanta Bachan, Adam Mickiewicz University, Poland
- Joseph Mariani, LIMSI-CNRS, France
- Jyoti D. Pawar, Goa University, India
- Kalika Bali, MSRI, Bangalore, India
- Kevin Scannell, St. Louis University, USA
- Khalid Choukri, ELRA, France
- Lars Hellan, NTNU, Norway
- Malhar Kulkarni, IIT-Bombay, India
- Manji Bhadra, Bankura University, West Bengal, India
- Marko Tadic, Croatian Academy of Sciences and Arts, Croatia

- Massimo Monaglia, University of Florence, Italy
- Monojit Choudhary, MSRI Bangalore, India
- Narayan Choudhary, CIIL-Mysore, India
- Nicoletta Calzolari, ILC-CNR, Pisa, Italy
- Niladri Shekhar Dash, ISI Kolkata, India
- Panchanan Mohanty, University of Hyderabad, India
- Pinky Nainwani, Optimum Pvt.Ltd, Bangalore, India
- Pushpak Bhattacharya, Director, IIT-Patna, India
- Qun Liu, Adapt Center, Dublin City University, Ireland
- Ritesh Kumar, Dr. B.R. Ambedkar University, Agra, India
- Sachin Kumar, CDAC-Pune, India
- Shivaji Bandhopadhyay, Director, NIT Silchar, India
- Sobha L, AU-KBC Research Centre, Anna University, India
- S S Aggarwal, KIIT, Gurgaon, India
- Stelios Piperidis, ILSP, Greece
- Subhash Chandra, Delhi University, India
- Swaran Lata, Head TDIL, MCIT, Govt. of India
- Virach Sornlertlamvanich, Thammasat Univeristy, Bangkok, Thailand
- Vishal Goyal, Punjabi University Patiala, India
- Zygmunt Vetulani, Adam Mickiewicz University, Poland

Preface

WILDRE – the 4th Workshop on Indian Language Data: Resources and Evaluation is being organized in Miyazaki, Japan on 12th May, 2018 under the LREC platform. India has a huge linguistic diversity and has seen concerted efforts from the Indian government and industry towards developing language resources. European Language Resource Association (ELRA) and its associate organizations have been very active and successful in addressing the challenges and opportunities related to language resource creation and evaluation. It is therefore a great opportunity for resource creators of Indian languages to showcase their work on this platform and also to interact and learn from those involved in similar initiatives all over the world.

The broader objectives of the 4th WILDRE will be

- to map the status of Indian Language Resources
- to investigate challenges related to creating and sharing various levels of language resources
- to promote a dialogue between language resource developers and users
- to provide opportunity for researchers from India to collaborate with researchers from other parts of the world

The call for papers received a good response from the Indian language technology community. Out of twenty nine full papers received for review, we selected one papers for oral, four for short oral, seven for poster and three for demo presentation.

Workshop Programme

12th May 2018

09:00 – 09:45hrs: Inaugural session

09:00 – 09:05hrs – Welcome by Workshop Chairs
09:05 – 09:25hrs – Inaugural Address
09:25 – 09:45hrs – Keynote Lecture

09:45 – 10:30hrs – Panel discussion
Coordinator: **Zygmunt Vetulani**
Panellists ¯ TBD

10:30 – 11:00hrs – Coffee break + Poster/Demo
Chairperson: **Kalika Bali**

- Royal Jain, *Anger Detection in Social Media for Resource Scarce Languages*
- Aniketh Janardhan Reddy, Monica Adusumilli, Sai Kiranmai Gorla, Lalita Bhanu Murthy Neti and Aruna Malapati, *Named Entity Recognition for Telugu using LSTM-CRF*
- K. V. S. Prasad, Shafqat Mumtaz Virk, Miki Nishioka and C. A. G. Kaushik, *Crowd-sourced Technical Texts can help Revitalise Indian Languages*
- Debopam Das, *Discourse Segmentation in Bangla*
- Priya Rani, Atul Kr. Ojha and Girish Nath Jha, *Automatic Language Identification System for Hindi and Magahi*
- Abdul Basit and Ritesh Kumar, *Towards a Part-of-Speech Tagger for Awadhi : Corpus and Experiments*
- Rajneesh Pandey, Atul Kr. Ojha and Girish Nath Jha, *Demo of Sanskrit-Hindi SMT System*
- Srishti Singh and Girish Nath Jha, *Demo: Parts-of-Speech Tagger for Bhojpuri*
- Mayank Jain, Yogesh Dawer, Nandini Chauhan and Anjali Gupta, *Developing Resources for a Less Resourced Language: Braj Bhasha*
- Atul Kr. Ojha and Girish Nath Jha, *Demo: Graphic-based Statistical Machine Translator*

11:00 – 12:45hrs – Paper Session II (Oral and Short Oral Presentation)
Chairperson: **S. S. Aggarwal**

- Massimo Moneglia, Alessandro Panunzi and Lorenzo Gregori, "T*aking Events" in Hindi. A Case Study from the Annotation of Indian Languages in IMAGACT*
- Gaurav Mohanty, Pruthwik Mishra and Radhika Mamidi, *Kabithaa: An Annotated Corpus of Odia Poems with Sentiment Polarity Information*
- Manas Jyoti Bora and Ritesh Kumar, *Automatic Word-level Identification of Language in Assamese – English – Hindi Code-mixed Data*
- Shagun Sinha and Girish Nath Jha, *Issues in Conversational Sanskrit to Bhojpuri MT*
- Ritesh Kumar, Bornini Lahiri, Deepak Alok, Atul Kr. Ojha, Mayank Jain, Abdul Basit and Yogesh Dawer, *Automatic Identification of Closely-related Indian Languages: Resources and Experiments*

12:45 – 12:55hrs – Valedictory Address

12:55 – 13:00hrs – Vote of Thanks

Table of Contents

Anger Detection in Social Media for Resource Scarce Languages

Royal Jain
Seernet Technologies LLC
royal.jain@seernet.io

Abstract

Emotion Detection from text is a recent field of research that is closely related to Sentiment Analysis. Emotion Analysis aims to detect and recognize different types of feelings through the expression of texts, such as anger, disgust, fear, happiness, sadness, surprise etc. Identifying emotion information from social media, news articles and other user generated content has a lot of applications. Current techniques heavily depend on emotion and polarity lexicons; however, such lexicons are only available in few resource rich languages and this hinders the research for resource scarce languages. Also, social media texts in Indian languages have distinct features such as Romanization, code mixing, grammatical and spelling mistakes, which makes the task of classification even harder. This research addresses this task by training a deep learning architecture on large amount of data available on social media platforms like Twitter, using emojis as proxy for emotions. The model's performance is then evaluated on a manually annotated dataset. This work is focused on Hindi language but the techniques used are language agnostic and can be used for other languages as well.

Keywords: anger detection, Indian languages, resource scare languages, deep learning, transfer learning

1. Introduction

Due to the growth of internet, an unprecedented amount of user generated content is available. This huge amount of data has introduced several new challenges and opportunities in the research communities. One of them is identifying user emotions and subjectivity in the text. Emotion Detection and Recognition from text is a recent field of research that is closely related to Sentiment Analysis. Sentiment Analysis aims to detect positive, neutral, or negative feelings from text, whereas Emotion Analysis aims to detect and recognize types of feelings such as anger, disgust, fear, happiness, sadness, and surprise through the expression of texts. It has many applications in real world, e.g. companies rely heavily on people's perspective of their goods and services, bloggers and content generators want to know the opinion of their readers, since the mid-2000s, governments around the world are paying increasing attention to the happiness index, etc. Anger detection is a sub-task of emotion detection which focuses on identification of text representing anger emotion. Reliable anger detection can be very useful in various fields, e.g. automatic customer service chatbots can use it as a signal of when humans should take over, it can be used to detect mental stress in workplace environment, etc.

Like many other NLP tasks, the biggest obstacle in emotion detection is lack of labelled data in sufficient amount. Consequently, co-occurring emotional expressions have been used for distant supervision in social media sentiment analysis and related tasks to make the models learn useful text representations before modelling these tasks directly. The state-of-the-art approaches within social media sentiment analysis use positive/negative emoticons for training their models (Jan Deriu and Jaggi., 2016). Hashtags such as anger, joy, happytweet, ugh, yuck and fml have also been used similarly by mapping them into emotional categories for emotion analysis in previous research (Jan Deriu and Jaggi., 2012). Distant supervision on noisy labels often enables a model to obtain better performance on the target task. However, these pseudo-labels are noisy as they are not always a direct label of emotional content. For instance, a positive emoji may serve to disambiguate an ambiguous sentence or to complement an otherwise relatively negative statement. (FA Kunneman and van den Bosch, 2014) discusses a similar duality in the use of emotional hashtags such as nice and lame. Twitter is a rich source of emotional texts but using them directly is a challenge as often emojis are not correct in depicting the emotion associated with these texts. Nevertheless, based on empirical observation we believe that in general anger emojis are used more reliably than others, meaning that the number of false positives obtained, when using anger emojis as proxy for emotion, are less when compared to using emojis for other emotions. In other words, there are few cases where the user will use an anger emoji when he/she is feeling some other emotion. This paper poses the task of anger detection as a binary classification problem where one class represents anger and other emotions are represented by the second class. This research shows that using emojis as proxy for anger on large dataset results can result in appreciable performance on manually annotated dataset.

Classical machine learning algorithms like SVM, Logistic Regression have performed reasonably well in various text classification tasks however their principal drawback is that for best performance they require manual feature engineering. Optimal set of features can vary both across languages and across domains. Hence, building a classification system for a new language is a cumbersome process and may not be very effective for all languages. We need a model which can train effectively on any dataset, irrespective of its language or text characteristics, with minimal or no manual adjustments across datasets.

Deep learning models have achieved astonishing results in several fields like Speech Recognition and Computer Vision, and have shown promising results when used for several NLP tasks. A major benefit of using deep learning models is that they don't require lot of feature engineering and hence are suitable for building language agnostic techniques for anger detection. A lot of research has been done for text classification using different architectures such as Convolutional Neural Networks (Kim, 2014), LSTMs

for tweet classification (Xin Wang and Wang, 2015) and Recursive Deep Learning Models for Sentiment Analysis (Richard Socher and Potts, 2013). This paper presents a deep learning architecture which uses a Bidirectional LSTM layer followed by an attention layer. A major benefit of LSTMs is that we don't need to worry about new inputs destroying important information, and the vanishing gradient doesn't affect the information to be kept.
Related work is summarized in section 2. Training dataset collection, pre-processing and properties are described in section 3.1, and test dataset is introduced in section 3.2. Model architecture is described in section 4. Experiments and Results are shown in section 5 and the paper is concluded in section 6.

2. Related Work

Most of the available techniques for emotion and subjectivity analysis rely heavily upon emotion and polarity lexicons like Emobank (Buechel and Hahn, 2017), opinion lexicon (Hu and Liu, 2004) etc. However, creation of these resources is expensive and cumbersome process and hence not feasible for a large number of languages. Consequently, these type of language resources are available only for a few resource rich languages. Lot of work has been done to overcome this scarcity of resources mainly in the field sentiment analysis. Most of the approaches depend on translation between a resource scarce language and resource rich language, e.g. (Balahur and Marco Turchi, 2012) used Bing, Google and Moses translation systems for sentiment analysis in French, German and Spanish. (Balahur and Turchi, 2013) uses a English tweet sentiment classification system to classify tweets translated into English from Italian, Spanish, French and German. These works however have several drawbacks, the primary one being the unavailability of good machine translation system for large number of language pairs. Social media tweets in many resource scarce Indian languages bring in more challenges as they are composed of texts both in Roman and Devanagari scripts, contain code mixed sentences and are often accompanied by grammatical and spelling mistakes. We believe that the effectiveness of translation based system is limited due to these reasons.
In the field of opinion mining a small amount of work has been done in Indian languages. Amitava Das and Bandopadhya (Das and Bandyopadhyay, 2010a), developed sentiwordnet for Bengali language. They apply word level lexical-transfer technique to each entry in English SentiWordNet using an English-Bengali Dictionary to obtain a Bengali SentiWordNet. Das and Bandopadhya (Das and Bandyopadhyay, 2010b) devised further strategies to predict the sentiment of a word like using interactive game for annotation of words, using Bi-Lingual dictionary for English and Indian Languages to determine the polarity, they also use wordnet and use synonym and antonym relations, to determine the polarity. Joshi et al. (Joshi et al., 2010) proposed a fallback strategy for sentiment prediction in Hindi language. This strategy follows three approaches: Using in-language resources for sentiment prediction. If the data is not enough using machine translation to obtain resources from a resource rich languages and if translation is not accurate using senti-wordnet like resources to predict the sentiment.
Using emotional expressions as noisy labels in text to counter scarcity of labels is not a new idea (Read, 2005) (Alec Go and Huang, 2009). Originally, binarized emoticons were used as noisy labels, but later also hashtags and emojis have been used. Most of these works base their emotion classification on the theories of emotion such as Ekman's six basic emotions (Ekman, 1992) which divides the emotion space into six basic emotions namely anger, disgust, fear, happiness, sadness and surprise.
Deep learning NLP models require word representations (containing context information) as input. One way to achieve this is randomly initializing the word vectors and trusting the emotion classification model itself to learn the word representations, besides the network parameters. However, this requires a large annotated corpus, which is difficult to obtain in most languages. The other way is to train a suitable deep learning model on a raw corpus in that language and then use the obtained embeddings of these in-language words as input to the emotion classification model. The most widely used embeddings are GLove (Pennington et al., 2014) and Google's word-2-vec system (Mikolov et al., 2013). Here, word embeddings generated using Facebook's Fasttext system (Bojanowski et al., 2016) on Wikipedia Hindi dump have been used. The benefit of using Fasttext embeddings is that it uses character n-grams as input and so it can easily compute word vectors for out-of-vocabulary words resulting in decent word vectors for slightly misspelled words, which is quite often the case in social media texts.
Recurrent Neural Networks have gained much attention because of their superior ability to preserve sequence information over time. Unfortunately, a problem with RNNs having transition functions of this form is that during training, the components of the gradient vector can grow or decay exponentially over long sequences. This problem with exploding or vanishing gradients makes it difficult for the RNN model to learn long distance correlations in a sequence. Long short-term memory network (LSTM) was proposed by (Hochreiter and Schmidhuber, 1997) to specifically address this issue of learning long-term dependencies. The LSTM maintains a separate memory cell inside it that updates and exposes its content only when deemed necessary. (Zhou et al., 2016) introduced BLSTM with attention mechanism to automatically select features that have a decisive effect on classification. (Yang et al., 2016) introduced a hierarchical network with two levels of attention mechanisms for document classification, namely word attention and sentence attention. This paper also implements an attention-based Bidirectional LSTM model.

3. Dataset

3.1. Training Dataset

In many cases, emojis serve as a proxy for the emotional contents of a text. Social media sites contain large amounts of short texts with emojis that can be utilized as noisy labels for training. For this paper, the data has been collected from Twitter, but any dataset with emoji occurrences could

be used. Hindi language tweets in both Roman and Devanagari script have been used for training dataset. Proper tokenization is very important for generalization over new data. All tweets are pre-processed to replace URLs, numbers and usernames by placeholders. To be included in the training set, the tweet must contain at least one emoji which strongly signifies emotion.

Now, we define the mapping of emojis to emotions. We use the definition of emojis present in Unicode's dictionary of emojis [1] to select the emojis which represent anger. Then we define another set of emojis which represent positive sentiment using the same dictionary. We hypothesise that if a tweet truly represent anger emotion then it should not contain any positive sentiment emoji. Therefore, we consider tweets which contains anger emojis and don't have any emoji which depicts positive sentiment, as anger tweets. This is done to reduce the number of false positives as much as possible. However, this results in a skewed distribution over tweets with only a small fraction of total tweets representing anger as shown in table 1.

Emotion	Number of Samples
Anger	4487
Others(Happy, Sad, Fear etc.)	51447

Table 1: Samples in each class

We also observe that a large number of Hindi tweets are written in transliterated Roman script. We leave them unchanged and let the model learn these as separate tokens. We do not introduce transliteration as it can potentially add errors and also because transliteration systems are not available for many language pairs.

Number of Tokens	832409
Romanized Tokens	397348
Avg Length in words	14.18

Table 2: Properties of Training Dataset

3.2. Test Data

We collected a small dataset of tweets in Hindi. We want to evaluate the predictive power of the model against all emotions and not just positive sentiment tweets, hence we made sure that the test dataset contains tweets representing fear, sadness, disgust along with joy, surprise and anger. The dataset was manually annotated by three different annotators. In case of disagreement in labels, the majority class was taken as the final label. We measured inter annotator agreement score using Cohen's kappa measure and obtained a score of 0.782.

4. Model Architecture

This paper uses a variant of the Long Short-Term Memory (LSTM) model that has been successful at many NLP tasks (Hochreiter and Schmidhuber, 1997). Our model uses an

[1] https://unicode.org/emoji/charts/full-emoji-list.html

Sample Label	Sample Count
Anger	49
Others	171

Table 3: Test Data Distribution

embedding layer with random initialization to project each word into a vector space. A rectified linear unit activation function layer is used to enforce a constraint of each embedding dimension being greater than or equal to 0. To capture the context of each word, a bidirectional LSTM layer with 128 hidden units (64 in each direction) is used. Finally, an attention layer that takes all of these layers as input using skip-connections is used (see Figure 1 for an illustration). The attention mechanism lets the model decide the importance of each word for the prediction task by weighing them when constructing the representation of the text. The output of this attention layer is used as input to the final Softmax layer for classification. We also added dropout layers for better generalization (Hinton et al., 2012). The model is implemented using Keras (Chollet and others, 2015) (with Tensorflow (Abadi et al., 2015) as backend).

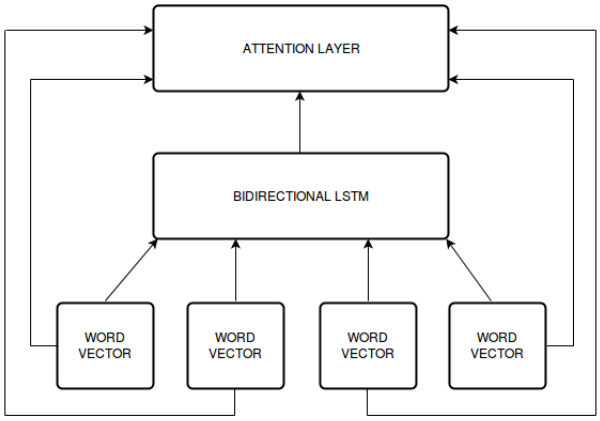

Figure 1: LSTM Model architecture

Another model is developed which has two channels for embedding layer. This is done to utilize the knowledge represented in pre-trained word vectors. First channel consists of an embedding layer which is randomly initialized and is updated along with model weights during training. This embedding layer is followed by a bidirectional LSTM layer which captures the context of each word and remembers the important features of input. Second channel has an embedding layer which is initialized by pre-trained word vectors obtained using Fasttext algorithm on Wikipedia Hindi dump. The second channel embedding layer is frozen and not updated during training. This second embedding layer is also followed by a bidirectional LSTM layer. Both of these Bidirectional LSTM layers and embedding layers are then concatenated and passed to the Attention layer. This is then followed by a softmax layer which makes the final classification decision. Model architecture is illustrated in Figure 2.

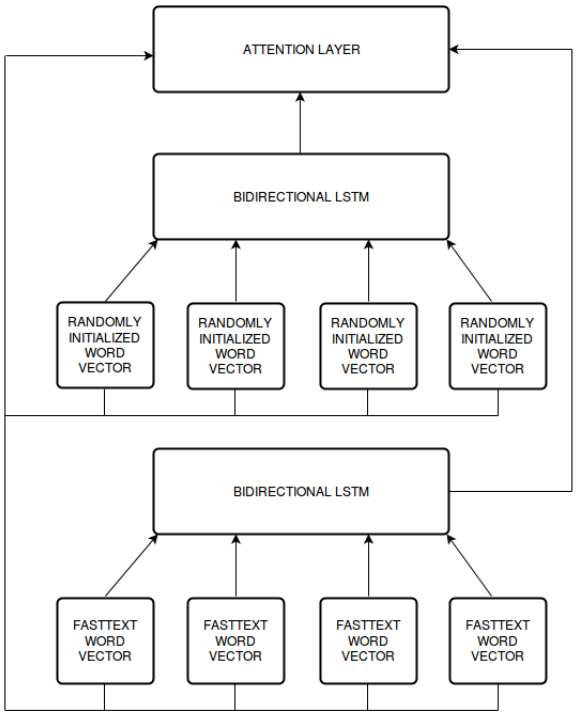

Figure 2: Multi-Channel LSTM architecture

5. Experiments and Results

5.1. Anger Detection

Our hypothesis is that training over large dataset of tweet data labelled using emojis, though noisy, will produce a good classifier for anger prediction. The performance of our models trained on automatically annotated datasets is evaluated on a manually annotated test corpus. F1 measure is used as primary metric since the data is skewed and hence accuracy would not be a strong measure of performance. We also report precision and recall score as in some cases false positives are considered more costly, in such cases we would prefer a model with better precision and where we need better coverage, a model with higher recall should be used.

Traditional classifiers over large input spaces, such as the the Bag of Words and Term frequency - Inverse Document Frequency (tf-idf) feature space, often provide strong baselines for text classification. We have used Naive Bayes and Logistic Regression in our experiments. Since the class distribution is highly skewed use change the class prior so that heavier loss is incurred when an anger data point is misclassified. As we observe from table 4, the results from these classifiers are very encouraging and support our hypothesis of using emojis as proxy for labels.

Model	F1	Precision	Recall
Naive Bayes	0.676	0.522	0.959
Logistic Regression	0.813	0.695	0.979

Table 4: Machine learning algorithm performance

We now compare best performing traditional machine learning classifiers with the deep learning architecture described in previous section. We use binary crossentropy as loss function with more weight given to data points of 'anger' class because of skewed data distribution. We split the dataset into training and validation set and the model is chosen based on the performance over validation set. 'Adam' (Kingma and Ba, 2014) algorithm is chosen for optimization. We use grid-search to choose model hyperparameters such as dropout, layer dimension etc. It can be seen in table 5, a significant improvement in classification performance over the traditional classifiers is observed.

Model	F1	Precision	Recall
LSTM	0.875	0.893	0.857
Multi-channel LSTM	0.889	0.88	0.897

Table 5: LSTM Model Performance

5.2. Model Analysis

We now analyze the predictions made by the model. Specifically we want to observe which emotions are more confusing for the model. Our hypothesis is that negative emotions are closer to each other and model can sometimes fail to differentiate between two negative emotions such as anger-sadness, fear-anger. From the false negatives we observe that majority of incorrect observations require knowledge of concept beyond the text, such as hostility between countries, hatred of TV shows etc. This also exemplifies the difficulty of detecting emotions and sentiment in general. While most of the false positives were those examples which didn't have clear majority in annotation, which shows even humans get confused whether a text represents anger emotion or is a comment on sad state of affairs.

6. Conclusion

In this work, it is established that, even without manually annotated dataset, anger can be detected in social media texts in a reliable fashion using a simple deep learning classification model, and easy usage of publicly available word embeddings for Hindi language. The main idea of the proposed approach is to develop a technique which can overcome the need of annotated resources. The experimental results show more than 88 percent F-score values. These observations substantiate the viability of the proposed approach in handling the key issues of multilingual emotion classification which are diversity of texts and scarcity of datasets across languages.

One major advantage of using deep learning is it's inherent ability to transfer knowledge to a different setting. Word embeddings are particularly useful as they learn the knowledge in context of a classification problem and can be directly used as input in a similar setting. In this context, words which denote anger should be, in some sense, near to each other. This means that similar task such as sentiment analysis might see an improvement using these embeddings as additional inputs.

The paper targets binary emotion classification for Hindi language. The paper shows that exploiting the embeddings and proxy labels over large dataset can result in appreciable performance over manually annotated dataset. This

work can further be extended to multi-class emotion classification problem which further distinguishes between other emotions such as sadness, disgust, joy etc. Also, further experiments can be conducted with other languages, publicly available resources, datasets and tools as well as other deep learning neural network configurations for emotion classification in resource scarce languages

7. Bibliographical References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Alec Go, R. B. and Huang, L. (2009). Twitter sentiment classification using distant supervision. cs224n project report, stanford.

Balahur, R. and Marco Turchi, . (2012). Multilingual sentiment analysis using machine translation?

Balahur, A. and Turchi, M. (2013). Improving sentiment analysis in twitter using multilingual machine translated data. In *In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2013, pages 49â55.*

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Buechel, S. and Hahn, U. (2017). Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Valencia, Spain, April 3-7, 2017. Volume 2, Short Papers, pages 578-585.*

Chollet, F. et al. (2015). Keras. `https://github.com/keras-team/keras`.

Das, A. and Bandyopadhyay, S. (2010a). Sentiwordnet for bangla.

Das, A. and Bandyopadhyay, S. (2010b). Sentiwordnet for indian languages. In *Proceedings of the Eighth Workshop on Asian Language Resouces*, pages 56–63, Beijing, China, August. Coling 2010 Organizing Committee.

Ekman, P. (1992). An argument for basic emotions. cognition and emotion, 6, 169-200.

FA Kunneman, C. L. and van den Bosch, A. (2014). The (un)predictability of emotional hashtags in twitter. In *In 52th Annual Meeting of the Association for Computational Linguistics (ACL). Association for Computational Linguistics.*

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Jan Deriu, Maurice Gonzenbach, F. U. A. L. V. D. L. and Jaggi., M. (2012). emotional tweets. In *In The First Joint Conference on Lexical and Computational Semantics (*SEM), pages 246â255. Association for Computational Linguistics*.

Jan Deriu, Maurice Gonzenbach, F. U. A. L. V. D. L. and Jaggi., M. (2016). Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of SemEval, pages 1124â1128.*

Joshi, A., Bhattacharyya, P., and R, B. (2010). A fall-back strategy for sentiment analysis in hindi: a case study, 12.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, ACLstudent '05, pages 43–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, J. Y. W. J. C. C. D. M. A. Y. N. and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *In Proceedings of EMNLP, volume 1631, pages 1631â1642*.

Xin Wang, Yuanchao Liu, C. S. B. W. and Wang, X. (2015). Predicting polarities of tweets by composing word embeddings with long short-term memory. In *In Proceedings of the 53rd Annual Meeting of ACL and the 7th International Joint Conference on Natural Language Processing, volume 1, pages 1343â1353*.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. J., and Hovy, E. H. (2016). Hierarchical attention networks for document classification. In *HLT-NAACL*.

Zhou, J., Cao, Y., Wang, X., Li, P., and Xu, W. (2016). Deep recurrent models with fast-forward connections for neural machine translation. *CoRR*, abs/1606.04199.

Named Entity Recognition for Telugu using LSTM-CRF

Aniketh Janardhan Reddy, Monica Adusumilli,
Sai Kiranmai Gorla, Lalita Bhanu Murthy Neti and Aruna Malapati
Birla Institute of Technology and Science, Pilani, Hyderabad, India
{f20140096,f20140005,p2013531,bhanu,arunam}@hyderabad.bits-pilani.ac.in

Abstract

Named Entity Recognition (NER) for Telugu is a challenging task due to the characteristic features of the language. Earlier studies have made use of handcrafted features and gazetteers for identifying named entities (NEs). In this paper, we present a Long Short-Term Memory (LSTM) - Conditional Random Fields (CRF) based approach that does not use any handcrafted features or gazetteers. The results are compared to those of traditional classifiers like support vector machines (SVMs) and CRFs. The LSTM-CRF classifier performs significantly better than both of them, achieving an F-measure of 85.13%.

Keywords: Named Entity Recognition, LSTM, CRF, Word Embeddings

1. Introduction

NER is an interesting task in Natural Language Processing (NLP) that identifies NEs such as the name of a person, location or organization in a sentence. NER has numerous applications in NLP and is used while performing text mining, machine translation, question answering, indexing for information retrieval, automatic summarization, etc.

Telugu is one of the most spoken Indian languages and is highly inflectional and agglutinating in nature. It has one of the richest and most complex set of linguistic rules resulting in complex word forms. The task of building an NER model for Telugu language has some linguistic challenges like the unavailability of annotated corpora, no gazetteer lists, no capitalization, spelling variations, free word order, etc. NEs in Telugu cannot be identified by capitalization as is the case with English and most European languages. Inflectional suffixes can be added either to the root or to the stem of Telugu words and common nouns can also be NEs in certain cases. The words are also more diverse in nature. NER in Telugu is challenging because of these difficulties. In this paper, we develop a classifier which performs NER in Telugu using LSTM and CRFs. We then compare the performance of our approach with those of popular tools like YamCha and CRF++. The annotated data used for our work has been made available for public use.

In Section 2. we discuss related work and in Section 3. we discuss our LSTM-CRF classifier. We describe the other classifiers we tested in Section 4. and our experiments, results and their analysis is documented in Section 5. Finally, we conclude in Section 6.

2. Related Work

State-of-the-art NER models are based on LSTMs. They overcome the problems associated with approaches that use handcrafted rules or gazetteers. These approaches are labor intensive and inflexible. Lample et al. (Lample et al., 2016) proposed a language independent LSTM-CRF based classifier which used pretrained word embeddings, character-level embeddings and contextual word representations. A CRF is finally used to perform classification. Our architecture is very similar to the one proposed by Lample et al. They also proposed another LSTM-based architecture inspired by shift-reduce parsers in the same paper. Chiu et al.(Chiu and Nichols, 2015) proposed a bidirectional LSTM (Bi-LSTM) and a Convolution Neural Network hybrid model that automatically detects word and character-level features. They reported an F-measure of 91.62% on the CoNLL-2003 dataset.

Considerable amount of work has been done on NER in other Indian languages such as Bengali and Hindi. Ekbal et al. (Ekbal and Bandyopadhyay, 2008) developed an NER model for Bengali and Hindi using SVMs. It uses various features which are computed based on both the word and the context in which it occurs. They reported F-measures of 84.15% and 77.17% for Bengali and Hindi respectively. In another paper(Ekbal and Bandyopadhyay, 2009), they proposed a CRF-based NER model for Bengali and Hindi and obtained F-measures of 83.89% for Bengali and 80.93% for Hindi. A CRF-based NER model which can handle nested NEs has been developed for Tamil(Vijayakrishna and Devi, 2008), another Dravidian language. Athavale et al.(Athavale et al., 2016) used a Bi-LSTM which took Word2Vec embeddings and the parts-of-speech (POS) tags of the tokens as input and output the NE tag. They reported an accuracy of 77.48% for NER in Hindi.

Srikanth and Murthy(Srikanth and Murthy, 2008) were some of the first authors to explore NER in Telugu. They built a two stage classifier which they tested using their own dataset. In the first stage, they used a CRF to identify nouns. Then, they developed a rule-based model to identify the NEs among the nouns. A CRF-based NER model was also built which made use of handcrafted features. Gazetteers were used to enhance the performance of their classifiers. Overall F-measures between 80% and 97% were reported in various experiments. It is interesting to note that the higher scores were obtained only upon using gazetteer lists. Their work also has several limitations. Firstly, they use handcrafted rules, features and gazetteers to perform NER which is both labor intensive and inflexible. Secondly, their classifier is only capable of identifying NEs which are one word long. Finally, neither their code nor their dataset has been made available publicly. Our work overcomes all of these problems. Shishtla et.al(Shishtla et al., 2008) built a CRF based NER model with language independent and dependent features and reported an F-measure of 44.89%.

3. The LSTM-CRF Classifier

We briefly introduce LSTMs, CRFs and other relevant concepts before proceeding to explain the architecture of the classifier we built.

3.1. Long Short-Term Memory (LSTM)

Recurrent Neural Networks (RNNs) are neural networks with self loops and are commonly used for building classifiers which operate on sequential data. In theory, vanilla RNNs are capable of learning long term dependencies between data points. But, it has been shown by Bengio et al. (Bengio et al., 1994) that they often fail to learn them due to vanishing gradients. LSTMs are a class of RNNs which were proposed by Hochreiter et al. (Hochreiter and Schmidhuber, 1997) to overcome this problem and are adept at learning long term dependencies. An LSTM takes sequential data of the form $(x_0, x_1, x_2...x_n)$ as input and gives a sequential output of the form $(y_0, y_1, y_2...y_n)$. LSTMs have an internal state which is updated whenever a new data point is input. This update is controlled by two gates. The forget gate determines how much of the previous state's information is to be retained. The input gate controls how much the state changes due to the new input. The LSTM finally gives an output which is determined by the output gate based on the internal state.

3.2. Character-Level Word Embeddings

The structure of a word is useful in determining if it is a named entity. For example, many Indian cities such as Hyderabad and Ahmadabad end with *-bad*. This structure can be captured through character-level word embeddings (Ling et al., 2015). Initially each character is assigned an n-dimensional vector. Each token is broken up into its individual characters which are then mapped to their corresponding vectors. Thus, a token is converted to a sequence of vectors which is then fed to a Bi-LSTM. A Bi-LSTM is composed of two LSTMs which process a sequence in opposite orders. The final states of both LSTMs are then concatenated to obtain the final character-level embedding of the token.

3.3. Generating Contextual Representations of Words using LSTMs

To make an output decision, an LSTM must store information about the previous data points. In the case of NER, the LSTM stores information about the tokens which occurred before the current token, thereby storing a contextual representation of the current token. In order to get both the left and right contexts we use a Bi-LSTM and then concatenate the internal states of the forward and backward LSTMs after processing the given token to get the token's final contextual representation.

3.4. Linear Chain Conditional Random Fields (CRFs)

When a conventional classifier is used to recognize NEs, each tagging decision occurs independent of the others. Entities can be spread across multiple tokens. For example, a person's name can be composed of two tokens consisting of his first and last names. There are also constraints on the occurrences of certain tags. For example, a sequence of I-xx tags cannot occur without a corresponding B-xx tag before them. It is also highly unlikely for entities to be present one after the other without separators between them. These observations make it clear that a tagging decision must be made after taking into account the tagging decisions for all the other tokens in the sequence.

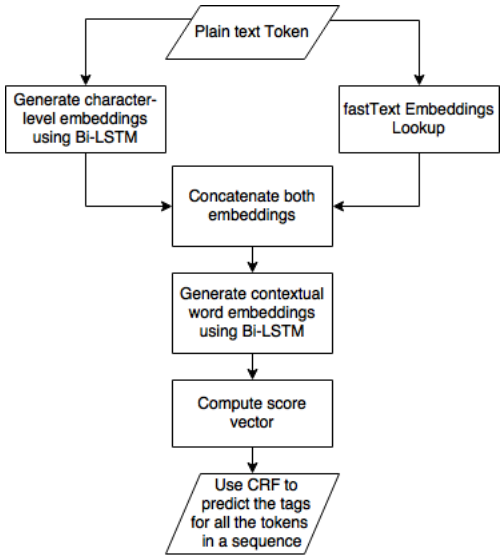

Figure 1: Main steps of our approach

Using CRFs (Lafferty, 2001), tagging is done for the entire sequence simultaneously and each tagging decision is dependent on the others. We use a linear chain CRF which considers dependencies between adjacent words (linear dependencies). Each token t_i in a sequence of the form $(t_0, t_1, t_2...t_n)$ has a corresponding m-dimensional vector s_i where m is the number of possible tags. Another $m \times m$ transition matrix A is used to capture the linear dependencies between the tagging decisions. A_{ij} of the matrix is indicative of the probability of the i^{th} tag being followed by the j^{th} tag. s and e are two m-dimensional vectors whose values represent the confidence of a tagging sequence starting and ending with a given tag respectively. Each tagging sequence of the form $(y_0, y_1, y_2...y_n)$ assigned to the token sequence is scored as:

$$Score(y_0, y_1, y_2...y_n) = s[y_0] + \sum_{i=0}^{n} s_i[y_i] + \sum_{i=0}^{n-1} A[y_i, y_{i+1}] + e[y_n]$$

The tagging sequence which has the maximum $Score$ is output by the CRF based classifier. While training, we seek to minimize the negative log of the probability of the correct tagging sequence $\tilde{Y}$. Hence our loss function is:

$$Loss = -log(P(\tilde{Y}))$$

where $P(\tilde{Y}) = \frac{e^{Score(\tilde{Y})}}{\sum_{y_0,...,y_n} e^{Score(y_0,...,y_n)}}$

During backpropagation, the various weights and embeddings of the model are tuned to minimize the $Loss$.

3.5. Neural Network Architecture

Figure 1 gives an overview of our architecture. Our classifier makes use of global word embeddings generated from raw text so as to capture the context of words in unseen text. In specific, we use the 300-dimensional fastText pretrained Telugu word embeddings provided by Facebook Research[1] (Bojanowski et al., 2016). We also generate 200-dimensional character-level embeddings for each token as described in Section 3.2.. The global word embeddings and the character-level embeddings are then concatenated for each token to represent them as vectors. Using the training data now represented in the form of sequences of vectors, 600-dimensional contextual word embeddings are generated for each token as described in 3.3.. The contextual word embeddings are then used to compute the scores for a word using a $600 \times m$ weight matrix W and an m dimensional bias vector b where m is the number of classes.

The score for each word is an m dimensional vector given by:
$s = W \cdot h + b$, where h is the contextual embedding of that word

After getting the scores for an entire sequence, we use a linear chain CRF to make the tagging decisions for the whole sequence as described in Section 3.4..
Our code and dataset is made publicly available for reproduction of results and future research[2]. Parts of the code were adapted from the code written by Guillaume Genthial[3].

4. Other Classifiers

We compare our classifier with two other language independent classifiers which are publicly available, YamCha[4] and CRF++[5]. This section gives a brief description of these tools. We tried out many configurations for these classifiers. Here, we describe the ones which performed the best.

4.1. YamCha

YamCha is an SVM-based open source toolkit which can perform many NLP tasks such as POS tagging, NER, base noun phrase chunking and text chunking. The features to be used and the parsing direction can be customized. In our study, we use YamCha with the following combination of static and dynamic features:

1. Current token and its POS tag
2. The two tokens preceding and the two tokens following the current token along with their POS tags.
3. NE tags of the two previous tokens

For each token, the aforementioned features are generated and supplied to YamCha for training. While testing, the tags assigned by the classifier to the previous two tokens are used as the last feature.

4.2. CRF++

CRF++ is an open source implementation of CRFs for segmenting or labeling sequential data. It is also language independent and like YamCha, it can be used for performing several NLP tasks. CRF++ is written in C++ and it uses the popular Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm to perform training. We used the following features in our study:

1. Unigram Features:
   (a) The current token, the two tokens which precede it and the two tokens which follow it.
   (b) Combination of the current token and the token before it.
   (c) Combination of the current token and the token after it.
   (d) The POS tags of the current token, the two tokens which precede it and the two tokens which follow it.
   (e) Combination of the POS tags of the two tokens which precede the current token.
   (f) Combination of the POS tags of the current token and the token which precedes it.
   (g) Combination of the POS tags of the current token and the token which follows it.
   (h) Combination of the POS tags of the two tokens which follow the current token.
   (i) Combination of the POS tags of the current token and the two tokens which precede it.
   (j) Combination of the POS tags of the current token, the token which precedes it and the token which follows it.
   (k) Combination of the POS tags of the current token and the two tokens which follow it.
2. Bigram Feature - Combination of the the current token and the NE tag of the previous token.

These features are generated for each input token and CRF++ uses them for training an NE tagger.

5. Experiments

5.1. The Corpus

We accumulated the Telugu text by crawling Telugu newspaper websites. It was then manually tagged by us. NEs belonging to four classes, namely, person (PERSON), location (LOC), organization (ORG) and other miscellaneous NEs (MISC) were manually identified and tagged using the standard IOB scheme. The data consists of 47966 tokens out of which 6260 are NEs and Table 2 shows the number of NEs belonging to each of the four classes. The tagset is given in Table 1 below.

[1] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md
[2] https://github.com/anikethjr/NER_Telugu
[3] https://github.com/guillaumegenthial/sequence_tagging
[4] http://chasen.org/~taku/software/yamcha
[5] https://taku910.github.io/crfpp/

Named Entity Tag	Example
PERSON	
	స్మృతి B-PERSON
	ఇరానీ I-PERSON
	(Smriti Irani)
LOC	
	సౌదీ B-LOC
	అరేబియా I-LOC
	(Saudi Arabia)
ORG	
	బిర్లా B-ORG
	ఇన్స్టిట్యూట్ I-ORG
	ఆఫ్ I-ORG
	టెక్నాలజీ I-ORG
	అండ్ I-ORG
	సైన్స్ I-ORG
	(Birla Institute of Technology and Science)
MISC	
	రూ. B-MISC
	35 I-MISC
	లక్షలు I-MISC
	(Rs. 35 lac)

Table 1: Named Entity Tagset

Class	Number of NEs
PERSON	1563
LOC	1915
ORG	778
MISC	2004
Total	6260

Table 2: Distribution of NEs

5.2. Training

Our classifier is trained using the Adam optimizer (Kingma and Ba, 2014) with the learning rate equal to 0.001 and the learning rate decay factor as 0.9. The batch size was equal to 20 and we also set a dropout rate of 0.5 while training. The training was carried out for 20 epochs.

5.3. Evaluation

10 sets of training and testing data were generated using the annotated corpus. 80% of the sentences present in the corpus comprise the training set and the remaining 20% comprise the test set. This split is done randomly and sentences are not repeated in the training and testing data. We then use these 10 splits to evaluate our classifier. The standard CoNLL evaluation script[6] is used to compute various evaluation metrics.

Approach	Precision	Recall	F-measure
YamCha	80.61	76.07	78.27
CRF++	80.75	74.92	77.72
LSTM-CRF	87.15	83.22	85.13

Table 3: Overall metrics for the various classifiers

Class	Yamcha	CRF++	LSTM-CRF
PERSON	76.61	76.08	80.07
LOC	81.21	80.52	88.11
ORG	54.32	55.86	68.02
MISC	82.10	84.56	92.68

Table 4: Comparison of the F-measures of the classifiers for each class of named entities

5.4. Results and Analysis

Tables 3 and 4 show the results we obtained. These metrics are computed after averaging the metrics over the 10 runs. The LSTM-CRF classifier clearly outperforms the other classifiers based on all the metrics. In fact, its performance is greater by approximately 7% based on overall F-measure. This is because of three main reasons. Firstly, the use of CRFs makes tagging more accurate because the classifier is able to discern the dependencies which exist between the individual tags. These dependencies can not be captured by an SVM, thereby lowering its performance. Secondly, character-level embeddings allow the classifier to learn the general form of an NE. Hence, our classifier is able to handle unknown NEs because it is able to identify them based on their structure. This cannot be accomplished when using tools like CRF++. Finally, Bi-LSTMs are great at learning long term dependencies and further augment the classifier's ability to learn dependencies between tokens which is not possible when using either SVMs or CRF++.

6. Conclusion

In this paper, we described an LSTM-CRF based approach for NER in Telugu. The approach makes use of pretrained fastText embeddings and character-level embeddings generated using a Bi-LSTM in order to learn contextual word embeddings using another Bi-LSTM. A linear chain CRF is finally used to perform the tagging based on the contextual word embeddings. The use of various word embeddings and a CRF allows the classifier to capture more contextual information and tagging dependencies respectively. The overall F-measure achieved using the LSTM-CRF classifier is approximately 7% greater than the F-measure obtained using SVM and CRF.

7. Bibliographical References

Athavale, V., Bharadwaj, S., Pamecha, M., Prabhu, A., and Shrivastava, M. (2016). Towards deep learning in hindi

[6]https://www.clips.uantwerpen.be/conll2003/ner/

ner: An approach to tackle the labelled data scarcity. *arXiv preprint arXiv:1610.09756*.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *Trans. Neur. Netw.*, 5(2):157–166, March.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Chiu, J. P. and Nichols, E. (2015). Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*.

Ekbal, A. and Bandyopadhyay, S. (2008). Bengali named entity recognition using support vector machine. In *IJCNLP*, pages 51–58.

Ekbal, A. and Bandyopadhyay, S. (2009). A conditional random field approach for named entity recognition in bengali and hindi. *Linguistic Issues in Language Technology*, 2(1):1–44.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Lafferty, J. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289. Morgan Kaufmann.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Ling, W., Dyer, C., Black, A. W., Trancoso, I., Fermandez, R., Amir, S., Marujo, L., and Luís, T. (2015). Finding function in form: Compositional character models for open vocabulary word representation. In *EMNLP*.

Shishtla, P. M., Gali, K., Pingali, P., and Varma, V. (2008). Experiments in telugu ner: A conditional random field approach. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

Srikanth, P. and Murthy, K. N. (2008). Named entity recognition for telugu. In *IJCNLP*, pages 41–50.

Vijayakrishna, R. and Devi, S. L. (2008). Domain focused named entity recognizer for tamil using conditional random fields. In *IJCNLP*, pages 59–66.

Crowd-sourced Technical Texts can help Revitalise Indian Languages

K. V. S. Prasad[1], Shafqat Mumtaz Virk[2], Miki Nishioka[3], C. A. G. Kaushik[4]
[1]Department of Computer Science and Engineering, Chalmers University of Technology, Sweden.
[2]Språkbanken, Department of Swedish, University of Gothenburg, Sweden.
[3]Graduate School of Language and Culture, Osaka University, Japan.
[4]Independent Researcher, Gothenburg, Sweden.
prasad@chalmers.se, shafqat.virk@svenska.gu.se, dumas@lang.osaka-u.ac.jp, kaushikcag@gmail.com

Abstract

Many Indian language (IL) speakers use English words for all STEM (Science, Technology, Engineering and Mathematics) concepts, however elementary, ignoring the STEM vocabulary in IL textbooks up to high-school. People assume English is necessary, and ILs are unfit, for STEM and higher education generally. English and STEM competence also mark wealth, so parents now abandon first language (L1) schools for often woeful "English" ones even at primary level. So children learn everything poorly: L1, English and content. To reverse this collapse, people need to use L1 more broadly. This paper calls for IL STEM texts, crowd-sourced from STEM-trained IL-speakers, to seed such usage. We note how the texts would fit in the linguistic landscape. They would also be important new data for computational linguistics. STEM-trained people with rusty L1 writing, like us, will find that with the dictionaries and text online, they can write in L1—we comment on vocabulary and help from related languages. Crowd-sourced texts vary in quality, but they can help people to use L1 for STEM topics, and to realise that children learn content better in L1 than in bad English.
Keywords: crowd-sourcing, STEM, Indian languages, translations

1. Background and Overview

The Multilingual FrameNet Project (MLFN, 2017) is using translations of the popular TED talk (Robinson, 2006) for cross-linguistic comparisons of specific uses of words. We were to work (Virk and Prasad, 2018) with the Hindi, Kannada, and Telugu translations, but these had not yet been made. We translated the talk ourselves, finding that with help from the dictionaries and IL text online, even our rusty IL writing skills could convey factual non-literary content.

That Indians can access this talk only through English is typical. There is little writing in Indian languages (ILs) about STEM (science, technology, engineering, mathematics, and related subjects). This diminishes ILs, and harms IL speakers, as we shall see.

A body of online IL STEM[1] texts, added to regularly, will help revive ILs. The texts can be crowd-sourced from STEM-trained IL speakers, an effective solution that will also build IL communities.

2. Indian languages today

ILs are used in daily conversation, songs, films, television, newspapers, and to talk about politics, religion and philosophy. IL stories, poems and translations of literary works are published, if mostly in small circulation journals. School texts up to the final year are available in ILs (NCERT, 2012; TN Govt., 2011) for most subjects including science and mathematics, as are glossaries from CSTT(CSTT, 2018).

But ILs provide few college texts in STEM, and English is needed to study at university or get a job (Sharma, 2017; The Hindu, 2016), or just to read or publish. Even in other fields, English is often the primary language, as a visit to any bookshop will confirm.

Governments impose official ILs on employees and students, with examinations to be passed for promotion. Why do ILs need to be imposed? Because they have little academic prestige and no employment value in many desirable fields: they use English terms to deal with the simplest technical matter. All over India, any conversation with a professional (doctor, engineer, lawyer) is peppered with English words. The school books and glossaries cited earlier provide the needed STEM words in ILs, but the public rarely uses them, and the resulting conviction that STEM can only be spoken about in English devalues ILs, and harms all IL based education and even thinking.

3. The costs of a devalued L1

We look at the cases of first language (L1) or English as the medium of instruction in school, and then at an L1 being actively dismantled by its own speakers.

3.1. First language (L1) medium

Some students in rural areas, e.g. in Karnataka, have good L1 and STEM skills, but avoid higher STEM courses for fear of the English medium these entail (Sharma, 2017). Poor English preparation has robbed these students of their preferred field of study.

Many students from Tamil medium schools do well at STEM and enter engineering colleges, but struggle there because their school English is poor(The Hindu, 2016). Even if their college lecturers speak to them in Tamil, the English terminology is daunting. There are few Tamil books or other sources at college level, so they can fail even at their favourite STEM subjects. If they pass, English still threatens their job prospects.

These cases are bad, but redeemable, since the students are actually well educated, with good L1 and STEM skills. All they need is (1) better English in high school, and (2) L1 STEM texts to help in college:

[1]We use STEM both as a noun, meaning the subjects, and as an adjective, meaning "pertaining to those subjects".

1. Remedial English and STEM English. This is a matter of organisation, the responsibility of the sending school, receiving college, and state government. There are open access resources to help.

2. Make L1 translations of key papers and parts of books, as well as original material and L1-English bilingual glossaries, freely available online. Thousands of STEM students and professionals speak ILs, so this material can be crowd-sourced. Accumulated over time, it will help successive batches of students. Some source material will be copyrighted, but much is under free licenses.

3.2. English medium

The outcome depends on the quality of English.

1. **Poor English: serious damage.** Having seen problems with L1 as medium, and the need for English to get jobs, many parents pay money they can ill afford to send their children to "English medium" schools. Some states might move much instruction to English (TOI Edit., 2017).

   But the results may be disastrous (Mody, 2017), because English is spoken at best very badly by many teachers. Bilingual instruction is not possible: "teachers lack proficiency, never mind fluency, in English. So, their classrooms are not bilingual, just badly mixed up" (Mody, 2017). The result is that children learn nothing: not English, not L1, and not the subjects being taught. Hopefully, the children retain at least spoken L1, from which recovery can start.

2. **Good English: success and alienation.** Good English medium instruction from childhood will remain something for the few. These often do well with education and jobs, but other problems lurk. For all but a very few, Indian English (IE) is restricted to prose, with poetry and song being the preserve of ILs. So IE is not a full language. Those who speak IE do best if they have a full L1, with its literature and songs, perhaps even able to talk about their work in L1 with minimum code-switching. Otherwise, they might end up culturally impoverished in both L1 and English.

3.3. English devalues L1 by invasion

The rush to English can take absurd forms even outside education, with L1 speakers making their own language more "English", and actively damaging it.

Telugu speech now replaces many Telugu words by English ones. So "blood" in Telugu is no longer `raktaṃ` but `blaḍ`. Other examples among many are: Mummy, Daddy, book, books (the plural is separately imported), water, rice, oil, door, cotton, life, food, dog, cat, moon, soul, body, week, month, daily, against, common, open, enquire, and all numbers. Mostly nouns, but no part of speech is immune.

These imports are not like "bus","radio" or "telephone" which arrived along with the object. Nor are they advanced terms with no Telugu equivalent; these are unnecessary replacements of basic words. They do not help in learning English (the imports adapt to Telugu phonology and grammar), but impoverish vocabulary and make Telugu word games and songs harder.

Telugu once took `rakta` "of blood" and `pōṭu` "thrust", and made the transparent `raktapōṭu` "blood pressure". Now it has "b.p.", an opaque name. Since Telugu "blood" is now `blaḍ`, the loss of `raktapōṭu` is itself no longer a loss of transparency (perhaps this kind of loss should be named the *raktapotu syndrome*).

4. Current efforts have failed ILs

We believe a high-functioning L1 is essential to a person's cognitive, social and psychological well-being. Everyone needs English for international contacts in STEM and for business, but most Europeans, for instance, use second-language English for this, while at home doing everything in L1, including talking to doctors, mechanics, lawyers and bankers.

But seventy years after independence, India is more dependent on English than ever, and does not have this privilege. No Indian language, most of which are larger than most European languages, can manage STEM without English words, and large numbers of people are shut out of participating in STEM activities (Sharma, 2017). We think India's language policies have failed, utterly—they have produced the current situation. The example below should give pause.

Over-centralisation. The 2017 ASER report (ASER, 2017) notes in passing how far language centralisation has gone. They found that oral rehydration solution (O.R.S.) packets are available in only Hindi and English—across all states in India. (To assess whether people could read and understand written instructions, the O.R.S. text was created in all 13 languages of the ASER survey). We can but repeat: to enable development, and protect life, respect L1.

Books are not enough. Textbooks, and some popular books and magazines on STEM topics are available in ILs. Government has tried to take STEM to the people in their own languages[2], and individuals have published popular science books in, for example, Telugu and Kannada (Verne, 2017; Vemuri, 2017; Hegde, 2017). Why are these efforts not enough?

Because without follow-up, books and magazines leave you high and dry. Also, no fixed set of texts can meet all needs. If you're going to Italy, a book on China does not help. English readers simply look up Italy in Wikipedia or other regularly updated websites. We would like to give IL readers the same luxury.

5. Clarification of our goals

The long-term goal is a high-functioning L1 in everyday speech, overturning the conviction that the sim-

[2]E.g., Two popular science periodicals in Hindi(Pragati, 2017; Homi Bhabha Centre for Science Education, 2017) publish articles accessible after registering as a user and logging in. The regional languages are looked after by state governments, but do not have the same level of resources.

plest STEM matter is the exclusive preserve of English. For this, IL-STEM vocabulary has to be used, regularly, in L1 speech and writing. Speakers should be able to replace English words by L1 words unselfconsciously, both in conversations about technical matter, and in college STEM texts (these latter might occasionally have to invent new L1 STEM vocabulary).

The reward will be the extension of L1 immediacy of understanding and expression to STEM matters. Children can see STEM being discussed in L1, both in real life and on social media, without reaching quite so often for an English word. The L1 words from their science and mathematics books will become real, and the word building and flow of L1 will work here too. With good L1, and STEM acquired through L1, the addition of English will become a less critical second language matter.

These goals are but dreams just now. As seeds, and continuing support for such development of ILs, we suggest crowd-sourced open access articles.

6. Seed: crowd-sourced STEM texts

Since crowd-sourcing is *crowd* based, it can produce the articles and websites mentioned above, in various languages, independent of government action or policy other than access to the internet. Just as cable television opened many different language channels, the internet can be used to revive ILs. People can now improve their own languages. How well any language group does will depend on how well their STEM trained speakers respond to this need.

We now outline a plan to build a body of STEM writing in L1, and say why it is feasible.

Hosting. The host for the contributed texts should provide minimal supervision and editing, but require proper acknowledgment of sources. It could focus on a particular L1 and provide aids for writers: links to online dictionaries, wiktionaries and other resources (several noted below), and previous contributions. It could provide tools for L1 such as parsers, treebanks, and wordnets, and forums where subject experts can collaborate with L1 experts, or where students can request L1 texts on specific STEM topics.

Amateur writers as contributors. Contributors need only be fluent L1 speakers who can write about the subject at hand (in English), and are willing to try to do so in L1. It does not matter if they last wrote in L1 long ago, if at all, or if their first L1 writing is difficult and slow, and less than perfect. The goal is only to convey the content of their ideas or of the original text they are translating.

Low barriers to entry. Using only known contributors might offer better quality, but the first requirement here is to get going. Style will come with practice, and texts can be improved.

The point is to encourage as many writers as possible, of both original texts and translations. The latter help build parallel corpora and provide texts outside the scope of the writer's immediate expertise.

Won't machine translation (MT) help? Yes, but first we need STEM texts in ILs. Statistical MT (SMT), such as used in Google's translation system, needs parallel corpora; with interest in IL technical texts decreasing, as it is now, there will be fewer texts for SMT to feed on, and it will be a self-fulfilling prophecy that MT fills social needs. Our goal of strengthening ILs breaks this downward spiral.

SMT is impressive, but less so for Telugu than for Hindi, for instance, presumably because there is less parallel corpus data for Telugu than for Hindi. So the crowd-sourcing we call for will be useful even if only to improve SMT systems[3].

It does appear that even the best MT systems still need a human to clean up the output, strengthening our argument that MT is only a support for languages people actually care about.

7. Notes on our novice L1 writings

We are IL speakers who write about STEM in English. Our L1 writing skills are rusty, and we have not read much L1 literature. In our L1 writing, we hope only to convey content. So we ourselves fit the minimum profile of the contributors we seek.

We found that with the dictionaries and text now online, amateur IL writing is quite feasible. This is anecdotal, but will hopefully encourage enough IL STEM contributions to generate more systematic data.

Notation. We use standard notation for the IL sounds. The palatalised spirant is written š. Retroflexion is shown by a dot under the letter; ṛ, a flap, is limited to Hindi and Urdu; ṇ and ṣ to Sanskritised words; and ḷ to Telugu and Kannada. Aspirated stops are shown thus: kʰ. A macron over a vowel denotes a long vowel, and ~, nasalisation. In Hindi and Urdu, e and o are always long, so the macron is dropped. ñ is the nasal homorganic with the following consonant.

7.1. Resources

Listed for Hindi, but apply similarly to other ILs.

Dictionaries. Wiktionary(Wiktionary, 2017b) uses English as the meta-language[4] and offers etymologies: (Wiktionary, 2017a) has sub-pages for words borrowed from specific languages. Over a thousand Perso-Arabic content words are listed for Hindi; Urdu would have more. This is a partial list: e.g., होशियार hošiyār "clever" is listed, but not होश hoš "consciousness".

Collins English-Hindi dictionary (Collins, 2017) is good, but limited in scope; the English example sentences are excellent, but there are no Hindi ones. Classic dictionaries are available from (Univ. of Chicago, 2017). Shabdkosh (Shabdkosh, 2017) is useful for throwing up many possible synonyms, but offers no contexts and examples to choose between them.

[3]The alternative to SMT, rule based MT, goes back for ILs at least to the Akshar Bharati group (Akshar Bharati et al., 1996) since the early 1990's. The Anusaaraka translator (Anusaaraka, 2017) is a testament both to the quality of their work, and to the difficulty of rule based MT.

[4](Hindi-Wiktionary, 2017) has Hindi descriptions.

MT. Google's translate is becoming very good indeed, but still needs cleanup, and does not distinguish between forms of Hindi[5].

Urdu and Hindi "share the same grammar and most of the basic vocabulary of everyday speech" (Flagship, 2012; Prasad and Virk, 2012); and (Bhat et al., 2016) says they are different standard registers or literary styles of the same language. So translation between the two should be shallow transfer, involving almost only lexical substitution. Apertium (Apertium, 2017) is a tool that does such jobs. Our Hindi translation of the Robinson talk begin by feeding an existing Urdu translation (Hassan and Anjum, 2006) into Apertium, and manually cleaning up the output.

7.2. Cultural connotations

The problems of translating across "unbridgeable cultural differences" are dealt with by (Prattipati, 2017) in the context of Bible translation to Telugu. Perhaps surprisingly, such problems can appear in technical matters too.

Mathematics has cultural foundations as (Raju, 2007) points out. "Proof" is translated to Hindi as प्रमाण `pramāṇ`, but in mathematics the former nowadays means logical deduction, whereas "Indian philosophy considered empirical proof `pratyakṣa` as more reliable than logical inference" (Raju, 2013). So the translation may have very different connotations, particularly for the reader in touch with their linguistic roots.

Sanskrit for STEM? Following on from that last remark, we note that there seem to be few STEM-trained people who can read the Sanskrit scientific and mathematical literature, and it seems to us that one long term goal is to build up such a community. Whether or not one agrees with Pollock in most of his paper (Pollock, 2011), it does seem that he is right that colonialism did "[...] render the literary past unreadable to most Indians".

7.3. Telugu and Kannada

Agglutination. Telugu and Kannada are agglutinative languages, so it is easy to produce a word that is in no dictionary. The question for the unsure writer is how to check that the word is acceptable. The first step is to search for the new word and see if it appears in other texts on the web. If it does occur, it is important to judge if the source is trustworthy.

If the word is not found on the web, the next step is to search for variations. So if వచ్చినట్టున్నాడా `vaccinaṭṭunnāḍā` "does it look like he has come?" is not found, then search for `vaccinaṭṭunnāḍu` "it looks like he has come" or `koṭṭinaṭṭunnāḍu` "it looks like he has hit", etc., to confirm that a nearby structure is in use. Native speaker judgement can be trusted to say that if "it looks like he has come" appears, then the proposed "does it look like he has come?" is acceptable.

Vowel Harmony. This is a feature of Telugu but not of Kannada. It means that in Telugu it is sometimes not clear what vowel one is using in speech. So కఠిచి `karici` "having bitten" can also be written and spoken `karaci`. Again, the solution is to search for the various possibilities and see what turns up, which is more popular, and so on.

7.4. Related languages

Telugu and Kannada are closely related languages. A translation of a sentence from one to the other often preserves morpheme order. This suggests that these languages make a good candidate pair for Apertium.

A novice IL writer will find themselves learning more about their L1, and so this paper ventures to comment on this experience. A traditional way to strengthen L1 is to learn languages close to L1; e.g., it used to be that many Kannada speakers also learned either Telugu or Tamil—useful for, say, music lyrics. But now, the three language formula means neighbours can often communicate only via Hindi or English. Learning these, or indeed any other, languages is good, but losing languages close to L1 is not.

7.5. Sanskrit borrowings

Since Sanskrit is a very important vocabularly resource for most ILs, it is important to understand how it fits in with any given L1.

For example, Telugu and Kannada are full of Sanskrit words commonly used, with no air of formality. As in many other ILs, Sanskrit has long been digested. The same word can even appear in original and many assimilated forms. Telugu has `nidra` "sleep" as in Sanskrit, but also `niddara` and `nidura`. Kannada has both `mūrti` "form" and `mūruti`, and so on.

Both the original form and the assimilated ones sit comfortably in speech, and in word building. "Go to sleep" can be `nidrapō` or `nidurapō` in Telugu[6]. But note that in Hindi, निद्रा `nidrā` can make निद्रायमान `nidrāymān` "one who is asleep", but नींद `nīnd`, the assimilated form of the Sanskrit word, cannot.

Word-building. This is important in STEM texts. E.g., "add" can be translated into Hindi as जोड़ना `jōṛnā` or योग करना `yōg karnā`, but only the latter Sanskrit form can make योगात्मक `yōgātmak`, "additive".

8. Summary

We have seen that currently L1s in India are devalued, and English, however poor, is seen as the route to progress. The context of the paper is a dream of restoring ILs to the status of full languages, used daily also for STEM and other technical subjects, and thus paving the way to fuller individual development.

1. Our primary contribution is a call to crowd-source STEM articles in ILs to complement existing IL

[5] "Hindi" covers quite different dialects, including Hindustani and "shuddh" Hindi. It thus has multiple *forms* (Kachru, 2006), and Standard Hindi is hard to define.

[6] Note that both `nidrapō` and `raktapōṭu` combine a native Dravidian word with a Sanskrit word, illustrating the degree of comfort with Sanskrit.

textbooks and glossaries, and keep up with new theories and technologies. These articles can help revitalise ILs and allow their speakers to comfortably navigate new worlds of ideas instead of being forced to do so in a foreign language.

If the new texts are translations, they will also improve the performance of MT for their language.

2. As STEM-trained IL speakers with rusty IL writing skills, we found in our writing efforts that the dictionaries and texts already online helped us write in L1 fairly confidently. So we believe our call is workable. In Sec. 7, we noted several factors of which the novice IL writer is likely to experience parallels.

3. We suggest that the star-shaped landscape of ILs today (interstate communication only via Hindi or English) should be softened to allow people to learn the languages close to their L1, a traditional way to strengthen L1, and allow language networks develop more naturally. Life critical information should be available in L1.

   A related decentralising idea is to develop tools to develop translators, such as Apertium, and other tools to work with related languages (such as Telugu and Kannada).

4. To go deeper into ILs could mean rediscovering Indian understandings of concepts such as proof. It would be useful to have more STEM-trained people who can read the Sanskrit STEM literature.

Conclusion. Further possibilities will reveal themselves as we go along. Knowledge grows only by sharing. Thanks to poor language development, India has denied its people full development, and grossly underutilized its human intellectual capital. We think crowd-sourcing can help set this right. It is a quick, cheap and inclusive way of tapping into existing potential, improving both understanding and L1 skills, and building a community of L1 writers.

Acknowledgement. We thank the referees for their comments, which focussed the paper and helped us remove extraneous material.

9. Bibliographical References

Akshar Bharati, Chaitanya, V., and Sangal, R. (1996). *Natural language processing: A Paninian perspective*. New Delhi: Prentice Hall of India.

Anusaaraka. (2017). Anusaaraka home page. https://anusaaraka.iiit.ac.in.

Apertium. (2017). Apertium wiki main page. http://wiki.apertium.org/wiki/Main_Page.

ASER. (2017). Annual Status of Education Report. www.asercentre.org.

Bhat, R. A., Bhat, I. A., Jain, N., and Sharma, D. M. (2016). A House United: Bridging the Script and Lexical Barrier between Hindi and Urdu. In *COLING*, pages 397–408. ACL.

Collins. (2017). English-Hindi dictionary. https://www.collinsdictionary.com/dictionary/english-hindi.

CSTT. (2018). Commission for Scientific and Technical Terminology, Main Page.

Flagship. (2012). *Undergraduate program and resource center for Hindi-Urdu at the University of Texas at Austin.* http://hindiurduflagship.org/about/two-languages-or-one/.

Hassan, S. and Anjum, U. (2006). Urdu translation of: Ken Robinson, Do schools kill creativity? TED profiles 133520 and 934090.

Hegde, N. (2017). Speaking science from the Kannada stage. http://www.thehindu.com/books/speaking-science-from-the-kannada-stage/article19974531.ece.

Hindi-Wiktionary. (2017). hindi vikshanari. https://hi.wiktionary.org/wiki/ .

Homi Bhabha Centre for Science Education. (2017). homi bhabha vijnan shiksha kendra. http://ehindi.hbcse.tifr.res.in.

Kachru, Y. (2006). *Hindi*. Philadelphia: John Benjamins Publ. Co. London Oriental and African Language Library.

MLFN. (2017). Multilingual FrameNet Project. framenet.icsi.berkeley.edu.

Mody, A. (2017). Lost for words: Why Tamil Nadu's shift to English medium instruction is not helping children. *scroll.in*. 17 Aug 2017.

NCERT. (2012). *School textbooks*. New Delhi: National Council for Educational Research and Training.

Pollock, S. (2011). Crisis in the Classics. *Social Research: An International Quarterly*, 78(1):21--48. India's World, a special issue. www.socres.org.

Pragati, V. (2017). vijnan pragati. http://www.niscair.res.in/sciencecommunication/Popularization%20of%20Science/vigyan0.asp.

Prasad, K. V. S. and Virk, S. (2012). Computational evidence that Hindi and Urdu share a grammar but not the lexicon. In *3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP), collocated with COLING 12*.

Prattipati, M. (2017). The Holy Register: Its Equivalence and Strategies. *Translation Today*, 11(1). http://www.ntm.org.in/download/ttvol/volume11-1/Art_4.pdf.

Raju, C. K. (2007). *Cultural Foundations of Mathematics*. Pearson Longman.

Raju, C. K. (2013). Teaching mathematics with a different philosophy, part 1: Formal mathematics as biased metaphysics. https://arxiv.org/ftp/arxiv/papers/1312/1312.2099.pdf.

Robinson, K. (2006). Do schools kill creativity? *TED: Ideas worth spreading*.

www.ted.com/talks/ken_robinson_says_schools_kill_creativity/up-next.

Shabdkosh. (2017). English to Hindi/Telugu /Kannada Dictionary. shabdkosh.com.

Sharma, A. S. K. V. S. (2017). *Challenges of Communicating Science in Regional Languages: Experiments in Kannada.* Springer. *Bridging the Communication Gap in Science and Technology*, Pallava Bagla and V. V. Binoy (eds.).

The Hindu. (2016). Language barrier puts students from Tamil medium in a fix. *Chennai: The Hindu, 13 July 2016*, July. Vasudha Venugopal and B. Aravind Kumar.

TN Govt. (2011). *School textbooks (Telugu and Kannada)*. Government of Tamilnadu, Department of School Education.

TOI Edit. (2017). Reform schools: Andhra Pradesh government's bold step in making English medium compulsory heralds change. *Times of India, 20 January 2017.*

Univ. of Chicago. (2017). Digital South Asia Library. dsalsrv02.uchicago.edu/dictionaries/.

Vemuri, V. (2017). Emi endukani. http://web.cs.ucdavis.edu/~vemuri/TeluguBooks.htm.

Verne, J. (2017). patalaniki prayanam. Telugu translation by Vaddadi Srinivas Chakravarthy of "Journey to the Centre of the Earth".

Virk, S. M. and Prasad, K. V. S. (2018). Towards Hindi/Urdu FrameNets via the Multilingual FrameNet. In *Proceedings of the International FrameNet Workshop (IFNW)*. To appear.

Wiktionary. (2017a). Category: Hindi terms by etymology. https://en.wiktionary.org/wiki/Category:Hindi_terms_by_etymology.

Wiktionary. (2017b). Wiktionary, the free dictionary. https://en.wiktionary.org/wiki/Wiktionary:Main_Page.

Discourse Segmentation in Bangla

Debopam Das
University of Potsdam
Karl-Liebknecht Strasse 24-25,
14476 Potsdam, Germany
debdas@uni-potsdam.de

Abstract

An important kind of discourse annotation is relational annotation in which texts are analyzed with respect to coherence relations (relations between text components, such as *Cause* or *Evidence*) present in the texts. Relational annotation according to Rhetorical Structure Theory (Mann and Thompson, 1988) typically begins with segmenting a text into minimal discourse units, which are then linked with each other (and later recursively with larger units) by certain coherence relations. As part of an ongoing corpus development project called the Bangla RST Discourse Treebank (Das and Stede, to appear), we have considered, examined and implemented a number of segmentation principles and strategies for dividing Bangla texts into minimal discourse units for the purpose of relational annotation. In this paper, we provide an overview of our annotation tasks, and describe our segmentation guidelines. We also present a few problems we encountered in segmenting Bangla texts, and discuss how we have addressed those issues.

Keywords: Bangla RST Discourse Treebank, discourse segmentation, Rhetorical Structure Theory, Bangla

1. Introduction

Relational annotation is a kind of discourse annotation that provides analysis of a text with respect to coherence relations (*Cause*, *Elaboration* or *Evidence*) that hold between the text components. Relational annotation tasks, according to Rhetorical Structure Theory or RST (Mann and Thompson, 1988), as followed in a number of RST-based discourse corpora, usually involves a number of sequential steps, typically beginning with the segmentation of texts into minimal discourse units. In RST, clauses are generally considered to be the basic units of discourse (Tofiloski et al., 2009). Nevertheless, RST segmentation policies differ from studies to studies, primarily because clauses are treated in different ways as information-bearing units, and partly because exceptions in the text data are handled in various manners.

We deal with segmentation of texts as part of an ongoing corpus development project called the Bangla RST Discourse Treebank or Bangla RST-DT (Das and Stede, to appear). This project builds a discourse corpus in Bangla which is annotated for coherence relations. RST-based corpora have been created for English (Carlson et al., 2002) and many other European languages, such as German (Stede, 2016), Dutch (van der Vliet et al., 2011), Brazilian Portuguese (Cardoso et al., 2011), Spanish (da Cunha et al., 2011) and Basque (Iruskieta et al., 2013). The practice has also been expanded to corpora in Asian languages such as Chinese (Cao et al., 2017) and Russian (Toldova et al., 2017), which are currently under production. We decide to contribute to this tradition by developing an RST corpus in Bangla, which, to our knowledge, is going to be the first dataset of its kind. As part of the relational annotation tasks, we have considered, examined and implemented a number of segmentation principles and strategies for dividing Bangla texts into minimal discourse units. In this paper, we present our segmentation guidelines, and discuss a few challenges associated with segmenting Bangla texts.

This paper is organized as follows: In Section 2., we provide a brief introduction of coherence relations and RST. Section 3. presents an overview of the Bangla RST-DT. In Section 4., we state the theoretical underpinnings of our segmentation guidelines, and describe different segmentation principles followed in the annotation. Section 5. presents a few issues in segmenting Bangla texts, and discusses how we have addressed them. Finally, Section 6. summarizes the paper, and provides the conclusion.

2. Coherence Relations and RST

The concept of coherence relations has been extensively studied in different discourse theories (see Das and Stede (to appear) for a list of theories and references), among which we chose to use Rhetorical Structure Theory or RST (Mann and Thompson, 1988) for our relational annotation purpose. This is because we believe that certain aspects of text organization are best captured by RST. We also chose RST because it is essentially a language neutral theory and it has been successfully used in many computational applications, such as text generation, discourse parsing, and text summarization (see Taboada and Mann (2006) for an overview).

Text organization in RST is described in terms of relations that hold between two or more non-overlapping text spans (discourse components). Relations can be multinuclear, reflecting a paratactic relationship, or nucleus-satellite, a hypotactic type of relation. The names nucleus and satellite refer to the relative importance of each of the relation components. Relation inventories are open, but the most common ones include names such as *Cause, Concession, Condition, Elaboration, Result* or *Summary*.

Texts, according to RST, consist of basic discourse units (also called elementary units or EDUs) that are connected to each other (or to larger units comprising two or more

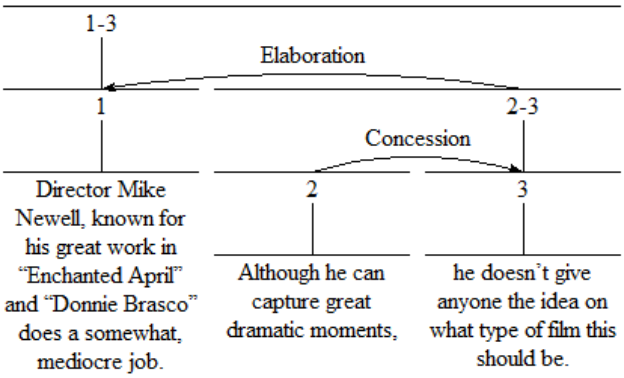

Figure 1: Graphical representation of an RST analysis

EDUs) by rhetorical (or coherence) relations in a recursive manner. According to Mann and Thompson (1988), the recursive application of different types of relations can be used to capture the entire structure of most texts. This, in practice, means that the RST analysis can be developed and represented as a tree structure in which the clausal units stand for the branches and the relations stand for the nodes.

For the purpose of illustration, we provide the annotation of a short text[1], represented by the tree diagram[2] in Figure 1. The text is segmented for three EDUs (minimal spans), which are marked by the cardinal numbers 1, 2 and 3, respectively. In the diagram, the arrow points to a span called the nucleus, and away from another span called the satellite. Span 2 (satellite) is connected to Span 3 (nucleus) by a *Concession* relation, and together they make the combined Span 2-3, which is further linked as a satellite to Span 1 (nucleus) by an *Elaboration* relation.

3. Bangla RST Discourse Treebank

Bangla RST-DT (Das and Stede, to appear) is a corpus of Bangla (currently under production) which is annotated for coherence relations following RST. The corpus contains 266 texts, comprising 71,009 words, with an average of 267 words per text. The corpus represents the newspaper genre. The texts have been collected from a popular Bangla daily called *Anandabazar Patrika* published in India. The texts in the corpus come from eight different sub-genres: (1) business-related news, (2) editorial columns, (3) international affairs, (4) cityscape (stories on Kolkata, the home city of the newspaper), (5) letters to the editor, (6) articles on nature, (7) features on science, and (8) reports on sports.

The annotation guidelines followed in the corpus[3] are based on the guidelines previously used in the Potsdam Commentary Corpus or PCC (Stede, 2016)[4], and are more closely related to an updated version of the PCC guidelines used in (Das et al., 2017)[5]. The corpus employs a set of 31 RST relations (26 mononuclear and 5 multinuclear relations), which are further divided in three groups: semantic, pragmatic and textual relations.

The Bangla RST-DT started with the annotation of 16 texts, taking two texts from each of the eight sub-genres mentioned above. The texts were pre-segmented by an expert annotator (the author of the present paper), and then they were separately annotated by three (one expert and two trained) annotators who are all native speakers of Bangla. The annotations were evaluated for inter-annotator agreement, with respect to span determination, nuclearity status assignation and relation labeling. The scores showed fairly high level of agreement between annotators, which indicates that our annotations are reliable. The currently-ongoing work includes the annotation of the remaining 250 texts, and we expect to complete the production of the corpus within the next few years. For more information about the corpus, see Das and Stede (to appear).

4. Segmentation in Bangla RST-DT

RST-based discourse segmentation strategies have been implemented (although with a moderate range of variation) by many previous studies for different languages, such as English (Tofiloski et al., 2009; Carlson and Marcu, 2001), German (Lüngen et al., 2006; Sidarenka et al., 2015), Brazilian Portuguese (Pardo and Nunes, 2008), Dutch (Abelen et al., 1993; den Ouden et al., 1998; van der Vliet et al., 2011) and Basque (Iruskieta et al., 2013).

The segmentation guidelines followed in the Bangla RST-DT are based on the guidelines used for German texts in the Potsdam Commentary Corpus or PCC (Stede, 2016) and for English texts in SLSeg (syntactic and lexically based discourse segmenter) (Tofiloski et al., 2009). Both PCC and SLSeg guidelines closely adhere to the original definition of spans in RST, according to which clauses constitute EDUs containing a verb, either finite or non-finite. More particularly, only adjunct, and not complement clauses, form legitimate EDUs. Broadly, coordinated clauses (but not coordinated verb phrases), adjunct clauses and non-restrictive relative clauses are considered as EDUs.

As we primarily follow formal criteria for determining the status of EDUs, we closely examine how clausal structures are realized in Bangla. For this purpose, we look into the existing literature on the Bangla grammar, and consult some notable works such as Chatterji (1988), Chakraborty (1992), Chaki (1996) and Sarkar (2006), which altogether provide a comprehensive account of clausal constructions in Bangla.

[1] Text source: SFU Review Corpus (Taboada, 2008)

[2] The RST diagram is created by RSTTool (O'Donnell, 2000) which provides a graphical representation of the RST analysis of a text in the form of a tree diagram. The tool is also used for doing the annotations in the Bangla RST-DT.

[3] http://angcl.ling.uni-potsdam.de/pdfs/Bangla-RST-DT-Annotation-Guidelines.pdf

[4] http://angcl.ling.uni-potsdam.de/resources/pcc.html

[5] http://www.sfu.ca/~mtaboada/docs/research/RST_Annotation_Guidelines.pdf

Although our segmentation guidelines are primarily meant to facilitate the annotation process in the Bangla RST-DT, the broader goal is to provide a set of RST-based discourse segmentation principles for Bangla, which can also be used for other Indo-Aryan languages, such as Assamese, Oriya or Punjabi. We believe that these guidelines can be adopted, modified and implemented according to specific annotation goals, and also that anyone having the basic knowledge of Bangla syntactic structures will be able to adequately follow them. Furthermore, since our segmentation principles mainly rely on formal criteria, they can also be used for the purpose of (semi-)automatic text segmentation, using the taggers and parsers available for Bangla (Hoque and Seddiqui, 2015; Ekbal and Bandyopadhyay, 2008; Hasan et al., 2010; Ghosh et al., 2009).

In the following subsection, we enumerate specific guidelines used for segmenting texts in the Bangla RST-DT. Most of the examples (accompanying specific guidelines) are taken from the corpus. The example sources (file numbers) are mentioned at the end of each example. If there is no file number, then the example is an invented one. The text within a pair of square brackets denotes an EDU. The text in the Bangla examples is written in the Roman script (ITRANS style).

4.1. Segmentation guidelines for Bangla

4.1.1. Zero-copula Constructions

Bangla allows frequent uses of zero-copula constructions, in which the main copular verb (corresponding to the verb 'be' or 'have' in English) remains absent on the surface, but in effect, is implied. Although in RST segmentation, a legitimate EDU is required to contain a verb, we decide to consider zero-copula constructions as clauses (headed by an implicit, but implied verb) and hence as EDUs, unless they act as complement clauses of other verbs.

(1) [sAjid o pArbhin svAmI-strI.]
Sajid and Parvin husband-wife
Sajid and Parvin are husband and wife. [kolkata-05]

4.1.2. Pro-drop Constructions

Bangla is a pro-drop language, in which subject pronouns are omitted from clauses on many occasions. In our annotation, we consider such (adjunct) clauses (clauses only with verbal predicates, and not the overt subjects) as EDUs.

(2) [er par Ar pratiyogitAmulak
this.Gen after anymore competitive
Asare nAmben nA.]
tournament will participate not
(He) will not participate in competitive tournaments anymore after this. [sports-03]

4.1.3. Clausal Subjects

Clausal subjects are not considered to be EDUs. In Bangla, clausal subjects are often manifested by verbal nouns.

(3) [upayukta sarkAri bandobasta thAkA
proper governmental provision be
jaruri.]
necessary
Having the proper governmental provision is necessary. [editorial-column-08]

Sometimes, a complete clause (with a finite verb) can also be used as the subject of a sentence.

(4) [se bandobasta ekebArei nei, emanTA
such provision at all.Emph not that
sambhabata balA yAbe nA.]
probably say can not
That there is no such provision at all cannot be said. [editorial-column-08]

4.1.4. Clausal Complements

Clausal complements include clausal objects of verbs, expressed as verbal nouns (Example 5) or infinitival clauses (Example 6), and they are not considered to form EDUs.

(5) [bahu mAnuSh dAktArer chembAre jAoYAr
many people doctors' to chamber go.Gen
cheYe jyotiShIr chembAre jAoYA beshi
than astrologers' to chamber go much
paChanda karen.]
prefer do
Many people prefer to go to astrologers' chambers than doctors' chambers. [letters-to-the-editor-06]

(6) [jiesTi kiChuTA hAsi phoTAte chaleChe
GST a little smile to bring go.Prog
bAik bhaktoder mukheo.]
motorcycle fans' face.Emph
GST is also going to bring a little smile on the faces motorcycle fans. [business-06]

4.1.5. Attribution Clauses

Attribution clauses are a kind of complement clauses, which are often represented by reported speeches, both directly (by direct quotes) or indirectly. We believe that attribution is a syntactic phenomenon, rather than a discourse one. Since attribution clauses act as the complements (more like noun clause complements) of the main reporting verbs in a matrix clause, they are not assigned the status of EDUs.

(7) [praphesar AYAn hoYAT boleChen, "ekhonai
professor Ian Howat said now.Emph
Ata.mkita haYe parar konao kAron nei."]
panicked be get.Gen any.Emph reason not
Professor Ian Howat said, "There is no reason to get panicked by now." [science-04]

(8) [goYendApradhAn Aro jAnAn,
the chief of detectives more informed
dhritader jiGYAsAbAd karA hochChe.]
arrested ones' interrogation do be.Prog
The chief of detectives also informed that the arrested ones are being interrogated. [kolkata-05]

Another way attribution clauses can manifest themselves is through cognitive predicates (containing verbs expressing feelings, thoughts or opinions, such as *think*, *know*, *estimate* or *wonder* in English). Just as in the case of reported speeches and for the similar reason, cognitive predicates are not treated as EDUs in our annotation.

(9) [hAmlAr prAthamik laxya t.NAr bA.Dii Chilo
of the attack primary target his house was
bale sandeha karChen tadantakArIrA.]
that suspicion do.Prog investigators
The investigators are suspecting that the primary target of the attack was his house. [international-01]

4.1.6. Relative Clauses

Relative clauses in Bangla are represented by correlative pronouns, sometimes in reduplicated forms (e.g., *ye / se, yini / tini, yata / tata, yArA yArA / tArA, yekhAne yekhAne / sekhAne sekhAne*). We exclude restrictive relative clauses from our consideration of EDUs.

(10) [jini lulAr sAjA ghoShanA karlen, tinio
who Lula's sentence announced he.Emph
rAjnItite Aste AgrahI.]
in politics to come interested
He who announced the sentence of Lula is also interested to join politics. [international-05]

However, non-restrictive clauses are considered to be EDUs in our annotation.

(11) [sirAj je mirjApharer upar bharsA koreChilen,]
Siraj that Mirzafar's on relied
[seTA pore tAr pataner kAran haYe
that later his downfall's reason be
d.NA.DAY]
stood
Siraj relied on Mirzafar, which later became the reason of his downfall.

4.1.7. Clauses with Correlative Discourse Connectives

In addition to correlative pronouns (for relative clauses), Bangla also contains correlative discourse connectives (sometimes in reduplicated forms) which are used to connect two clauses. Examples of correlative connectives include *ye hetu / se hetu, yeman (yeman) / teman (teman), yadi / tabe,* etc. Clauses with such connectives are considered to be EDUs in our annotation.

(12) [... hAmlA ye hetu tIrthayAtrIder upar,]
... the attack since the pilgrims.Gen on
[se hetu ei hAmlAr ek anYatara
that is why this of the attack a different
tAtparya kh.NojAr chestA hachChe.]
significance find.Gen attempt being
Since the attack was on the pilgrims, that's why there is being an attempt to find a different significance of the attack. [editorial-column-07]

4.1.8. Nominal Modifiers

Nominal modifiers represented by verbal nouns are not considered as EDUs. In Example 13, the noun *'bAsTike'* ('the bus') is modified by the verbal noun *'ulTo dik theke AsA'* ('coming from the opposite side') and hence, it is not segmented as an EDU.

(13) [ulTo dik theke AsA bAsTike dhAkkA mAre
opposite side from come the bus hit
oi gA.DiTi.]
that car
The car hit the bus coming from the opposite side. [international-01]

4.1.9. Participial Clauses

Participial clauses (with a past active participle), are considered to constitute legitimate EDUs.

(14) [dvitIYa TesTe phire ese] [sirij 1-1
second in the test coming back series 1-1
karlen phAph duplesi.]
did Faf du Plessis
Coming back in the second test, Faf du Plessis made the series 1-1. [sport-08]

4.1.10. Verbal Nouns with a Postposition

Verbal nouns, as already shown in Example 3 and 5, are not considered to be EDUs. However, when verbal nouns are used with a postposition, they are treated as EDUs. In Example 15, the verbal noun *'eman sambhAbanAder chine neoAr'* ('recognizing such potentials') with the postposition *'janya'* ('for') forms an EDU.

(15) [eman sambhAbanAder chine neoAr
such potentials recognize.Gen
janya] [upayukta sarkAri bandobasta thAkA
for proper governmental provision be
jaruri]
necessary
Having the proper governmental provision is necessary for recognizing such potentials. [editorial-column-08]

4.1.11. Infinitival Clauses

Infinitival clauses which are not complements of verbs are considered as EDUs.

(16) [nyAnoke bhabiShyate rAstAy chAlAte]
Nano in the future on road run
[dubaCharer madhyei chAi natun lagni.]
of two years within.Emph want new investments
New investments are required within the next two years in order to run Nano on road in the future. [business-05]

4.1.12. Conditional Clauses

Conditional clausal constructions in Bangla act like adjunct clauses, and hence they are considered to form EDUs.

(17) [jiesTir parimAn kam hale] [sexetre dAm
GST's amount small be.if then price
kambe gA.Dir]
will go down cars'

If the amount of GST is small, then the price of cars will go down. [business-06]

4.1.13. Coordinated Constructions

As in many other RST annotation studies, we also consider as EDUs only coordinated clauses (linked by a comma or discourse connective), but not coordinated verb phrases.

(18) [Aphsos karChilo bA.mlA,] [Aphsos karChilo mahAnagar.]
regret was doing Bangla regret was doing the big city
Bangla was regretting, so was the big city. [editorial-column-11]

In sum, we followed the basic ideas of RST segmentation from the PCC and SLSeg guidelines (for adjunct/complement clauses, attribution and relative clauses). However, at the same time, we have developed some new segmentation strategies suitable for certain Bangla constructions (e.g., conditional clauses). Sometimes, we used the existing PCC and SLSeg guidelines, but have adapted them in particular ways so that they comply with the syntactic and discourse structures of Bangla (in the treatment of relative clauses, verbal noun with a postposition, etc.).

5. Segmentation Issues and Resolutions

For us, the biggest challenge was to perform the RST segmentation for a non-European language, for which no previous documented effort on discourse segmentation was available. In particular, we have encountered a few issues in our segmentation task, which are described below:

1. Bangla employs the use of phrasal verbs, which (unlike in English) comprise a pre-verbal element and the main verb (which is marked for tense and person). In certain instances, we have noticed that the phrasal verb constructions and adjunct clause pairs have similar forms, and it is often difficult to distinguish them. For instance, Example 19 and 20 are very similar in form. However, in Example 19 the form *khete* is a pre-verbal element of the phrasal verb *khete baseChen*, while in Example 20 *khete* acts as an infinitival adjunct clause (with the implication "in order to eat") (cf. (Chakraborty, 1992), p. 137-138).

   (19) tini khete baseChen.
   he/she to eat sat down
   He/she sat down to eat.

   (20) tini khete geChen.
   he/she to eat went
   He/she went to eat.

   For this problem, we use a paraphrase test: We checked whether it is possible to replace the questionable item *khete* ('to eat') with *khAbAr janya* ('for eating' or 'in order to eat'), and if the modified construction still yields a grammatical output, then we consider it to be an adjunct clause (and hence an EDU). We used this test and other similar tests for resolving such ambiguities.

2. Some texts in our corpus contain long speeches (whether direct quotes or indirect reported speeches). According to our guidelines for attribution clauses, we do not segment between the reporting clause and the reported clause, or between the reported clauses. However, for longer speeches consisting of multiple sentences, we have observed that if we strictly follow this principle, we might end up losing significant information at the discourse level. Thus, we have decided to add an exception: If a reported speech (or quote) spans over more than one sentence, then each sentence will be segmented as EDUs (marked by square brackets in Example 21).

   (21) "[bhAloi haYeChe daurTA.] [Ami saThik pathei yAchCHi.] [tabe ekhanao anek kAj bAki.", baleChen bolT.]
   good.Emph has been the (sprint) race I right in-the-direction.Emph moving However still many things remaining said Bolt
   "The (sprint) race has been good. I am moving in the just right direction. However, there are still many things to do.", said Bolt. [sport-03]

3. Bangla makes use of correlatives (a pair of two particles) where one part presupposes the presence of the other. In the standard Bangla grammar (Chakraborty, 1992; Sarkar, 2006), correlatives provide a cover term for elements such as *yini / tini, yata / tata, ye hetu / se hetu, yeman / teman*, or *yadi / tabe* (see Section 4.1.6. and 4.1.7.). However, we have observed that these correlative elements have two distinct functions from a discourse point of view: Some correlatives (*yini / tini, yata / tata*, etc.) are used to establish coreferential relation between objects or entities, while others (*ye hetu / se hetu, yeman / teman, yadi / tabe*, etc.) are used for relating clauses or text spans. For this reason, we distinguish these two types in our annotation, and classify the former type as *correlative pronouns* (used in relative clauses) and the latter as *correlative discourse connectives* (used for linking clauses or text spans).

6. Conclusion

In this paper, we have presented the segmentation guidelines for annotating texts in the Bangla RST Discourse Treebank. We have discussed different segmentation principles and strategies, and motivated our reasons for choosing or developing those guidelines. Performing the segmentation for Bangla has also posed a few challenges for us, which we have successfully dealt with in our annotation task. We believe (as we have experienced) that in order to develop a set of RST segmentation guidelines in a new language one could adopt the basic segmentation principles from the available and recognized guidelines (such as the one for PCC or SLSeg), which could later be complemented by the language-specific guidelines or a modification of previous guidelines.

7. Acknowledgements

I would like to thank Dr. Manfred Stede for his active and sincere commitment to the Bangla Discourse Treebank project. Special thanks go to Dr. Pabitra Sarkar and Dr. Samir Sarkar for their invaluable suggestions on the topic of this paper.

8. Bibliographical References

Abelen, E., Redeker, G., and Thompson, S. (1993). The Rhetorical Structure of US-American and Dutch Fund-Raising Letters. *Text - Interdisciplinary Journal for the Study of Discourse*, 13(3):323–350.

Carlson, L. and Marcu, D. (2001). Discourse Tagging Manual. ISI Technical Report ISI-TR-545, University of Southern California.

Chaki, J. (1996). *Bangla Bhashar Byakaran*. Ananda Publishers, Kolkata, India.

Chakraborty, U. K. (1992). *Bangla Padaguccher Sangathan (Structure of Bengali Phrases)*. Dey's Publishing, Kolkata, India.

Chatterji, S. K. (1988). *Bhasha-Prakash Bangla Vyakaran*. Rupa and Company, New Delhi, India.

Das, D., Taboada, M., and Stede, M. (2017). The good, the bad, and the disagreement: Complex ground truth in rhetorical structure analysis. In *Proceedings of the 6th Workshop on Recent Advances in RST and Related Formalisms*, pages 11–19.

den Ouden, H. J. N., van Wijk, C. H., Terken, J. M., and Noordman, L. G. (1998). Reliability of discourse structure annotation. IPO Annual Progress Report.

Ekbal, A. and Bandyopadhyay, S. (2008). Part of Speech Tagging in Bengali Using Support Vector Machine. In *Proceedings of the 2008 International Conference on Information Technology*, pages 106–111.

Ghosh, A., Bhaskar, P., Das, A., and Bandyopadhyay, S. (2009). Dependency Parser for Bengali: the JU System at ICON 2009. In *NLP Tool Contest ICON 2009*, pages 87–91.

Hasan, K. M. A., Mondal, A., and Saha, A. (2010). A context free grammar and its predictive parser for Bangla grammar recognition. In *2010 13th International Conference on Computer and Information Technology (ICCIT)*, pages 87–91.

Hoque, M. N. and Seddiqui, M. H. (2015). Bangla Parts-of-Speech tagging using Bangla stemmer and rule based analyzer. *Proceedings of the 18th International Conference on Computer and Information Technology (ICCIT)*, pages 440–444.

Lüngen, H., Puskás, C., Bärenfänger, M., Hilbert, M., and Lobin, H., (2006). *Discourse Segmentation of German Written Texts*, pages 245–256. Springer Berlin Heidelberg, Berlin, Heidelberg.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

O'Donnell, M. (2000). RSTTool 2.4 – A markup tool for Rhetorical Structure Theory. In *Proceedings of the International Natural Language Generation Conference*, pages 253–256, Mizpe Ramon/Israel.

Pardo, T. A. S. and Nunes, M. d. G. V. (2008). On the Development and Evaluation of a Brazilian Portuguese Discourse Parser. *RITA*, 15:43–64.

Sarkar, P. (2006). *Bangla Byakaran Prasanga*. Dey's Publishing, Kolkata, India.

Sidarenka, U., Peldszus, A., and Stede, M. (2015). Discourse Segmentation of German Texts. *Journal for Language Technology and Computational Linguistics*, 30(1):71–98.

Taboada, M. and Mann, W. C. (2006). Applications of Rhetorical Structure Theory. *Discourse Studies*, 8(4):567–588.

Tofiloski, M., Julian, B., and Taboada, M. (2009). A Syntactic and Lexical-Based Discourse Segmenter. In *47th Annual Meeting of the Association for Computational Linguistics*, pages 77–80.

van der Vliet, N., Berzlánovich, I., Bouma, G., Egg, M., and Redeker, G. (2011). Building a discourse-annotated Dutch text corpus. In *Beyond Semantics, Bochumer Linguistische Arbeitsberichte 3*, pages 157–171.

9. Language Resource References

Cao, S., Xue, N., da Cunha, I., Iruskieta, M., and Wang, C. (2017). Discourse Segmentation for Building a RST Chinese Treebank. In *Proceedings of the 6th Workshop on Recent Advances in RST and Related Formalisms*, pages 73–81.

Cardoso, P., Maziero, E., Jorge, M. L. C., Seno, E., Di Felippo, A., Rino, L., Nunes, M. d. G., and Pardo, T. (2011). CSTNews - A Discourse-Annotated Corpus for Single and Multi-Document Summarization of News Texts in Brazilian Portuguese. In *Proceedings of the 3rd RST Brazilian Meeting*, pages 88–105.

Carlson, L., Marcu, D., and Okurowski, M. E. (2002). RST Discourse Treebank, ldc2002t07.

da Cunha, I., Torres-Moreno, J.-M., and Sierra, G. (2011). On the Development of the RST Spanish Treebank. In *Proceedings of the 5th Linguistic Annotation Workshop*, LAW V '11, pages 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Das, D. and Stede, M. (to appear). Developing the Bangla RST Discourse Treebank. In *Proceedings of the 6th Workshop on Recent Advances in RST and Related Formalisms*.

Iruskieta, M., Aranzabe, M. J., de Ilarraza, A. D., Gonzalez-Dios, I., Lersundi, M., and de Lacalle, O. L. (2013). The RST Basque Treebank: An online search interface to check rhetorical relations. In *Proceedings of the 4th workshop RST and discourse studies*, pages 40–49.

Stede, M. (2016). Rhetorische Struktur. In Manfred Stede, editor, *Handbuch Textannotation: Potsdamer Kommentarkorpus 2.0*. Universitätsverlag, Potsdam.

Taboada, M. (2008). SFU Review Corpus [corpus].

Toldova, S., Pisarevskaya, D., Ananyeva, M., Kobozeva, M., Nasedkin, A., Nikiforova, S., Pavlova, I., and Shelepov, A. (2017). Rhetorical relations markers in Russian RST Treebank. In *Proceedings of the 6th Workshop on Recent Advances in RST and Related Formalisms*, pages 29–33.

Automatic Language Identification System for Hindi and Magahi

Priya Rani, Atul Kr. Ojha, Girish Nath Jha
JNU, New Delhi
{pranijnu, shashwatup9k, girishjha}@gmail.com

Abstract

Language identification has become a prerequisite for all kinds of automated text processing systems. In this paper, we present a rule-based language identifier tool for two closely related Indo-Aryan languages: Hindi and Magahi. This system has currently achieved an accuracy of approx 86.34%. We hope to improve this in the future. Automatic identification of languages will be significant in the accuracy of output of Web Crawlers.

Keyword: Language identification, rule-based approach, Hindi and Magahi

1. Introduction

Code-mixing is a common phenomenon in countries like India where five different language families co-exist. According to a report issued by Microsoft Research, 95% of the languages used by Indians are mixed (Chittaranjan, 2014). This paper focuses on two very closely related Indo-Aryan languages: Hindi and Magahi. Hindi being a scheduled/official language (languages which are included in the 8th schedule of constitution of India.), is used for official purpose, spoken in north, western, central and eastern parts of India. Whereas, Magahi is a non-scheduled or non-official language spoken in eastern states of India including Patna, Gaya, Jehanabad, Munger, Begusarai, Hazaribagh, Nalanda districts of Bihar, Ranchi district in Jharkhand, some parts of Orissa and Malda district in West Bengal (Kumar, 2011). Even though due non-experts consider Magahi as one of the dialects of Hindi, linguists understand it as a separate language owing to significant difference between both the languages. According to Census 2001, Hindi is spoken by 534,271,550 people and Magahi speakers count up to 14,046,400.[1] In this paper, we report a rule-based language identifier tool for Hindi and Magahi. The immediate goal is to identify the language of a given text. The paper demonstrates the function, experimental set-up, efficiency and limitations of the tool.

1.1 Motivation of the Study

Language Identification is the process of finding the natural language in which the content of the text is encoded. (Garg ét al., 2014). It is an extensive research area used in various fields such as machine translation, information retrieval, summarization etc. It is easier to distinguish two languages belonging to different language families, and with different typological distributions. It becomes even more easier to distinguish two languages if they are encoded in different scripts. However, the identification task becomes challenging when the two languages belong to the same language family and share many typological and areal features. In this paper, we will develop a tool to identify two closely related languages Hindi and Magahi only that share many typological and areal features and belong to the same language family. Despite these relatedness, these languages differ from each other in many respect. We will focus on those differences and use them to develop the tool.

1.2 Features of Hindi and Magahi

The section deals with some basic linguistic features in an attempt to differentiate between Hindi and Magahi.

(a) A primary difference between both the languages is that while Magahi is a nominative-accusative language, Hindi is an ergative language.
For example:

Magahi

rəm-mɑ sit̪-wa ke əm-mɑ
ram-PRT sita-PRT to mango-PRT
delkai
givePST.
Translation- “Ram gave mango to Sita”

Hindi

ram-ne ʃit̪a-ko am dija
ram-ERG Sita-DAT mango give-PST

1https://www.ethnologue.com/language/hin and https://www.ethnologue.com/language/mag

Translation- “ram gave mango to sita.”

(b) Magahi, like other eastern Indo-Aryan languages and unlike Hindi do not show number and gender agreement. It reflects agreement with person and honorificity. Whereas Hindi shows agreement with phi features i.e person , number and gender as well as posses honorific agreement.
For example:

Magahi

i. sit̪-wa ɟa he
sita-PRT go AUX-3P.NH
“Sita is going.”

ii. apne ɟait̪ hatʰin
You.H go AUX-H
“you are going”

iii. həmni ɟa hi
we.NH go AUX-2P.NH
“we are going”

Hindi

iv. ʃit̪a ɟa rahi hai
sita go PROG.F AUX.3SG
“Sita is going”

v. aap ɟa rahe hain
You go PROG.H AUX.2SG.H
“you are going”

vi. həm log ɟa rahe hain
we all go PROG AUX.1PL
“we are going.”

(c) Numeral classifiers are prominent in Magahi but Hindi lacks them. For example:

Hindi	ek	ɖo	tınə
Magahi	e-**go**	ɖu-**go**	tın-**go**
Translation	one	two	three

(d) Nouns have two basic forms in Magahi : Base form and Inflected form. The particles -wa, -ia, -ma, -a are added to the base form to construct an inflected form.The nominal particles -ia, -a, -ma and -a are allomorphs of base form -wa. (Alok, 2010).These are used to show different linguistic features. These particles are addded to proper names as well. Whereas nouns in Hindi have only one form. For example:

Magahi

	Form1	Form2
i.	gʰər	gʰər-wɑ
	house	house-PRT
ii.	ɑm	əm-mɑ
	mango	mango-PRT
iii.	rɑm	rəm-mɑ
	Ram	Ram

Hindi

iv. gʰər
house

v. ɑm
mango

(e) Verbs shows some interesting and complex features in both languages. The difference lies in inflections that they take. Magahi present tense is unmarked, past tense is marked with‘-l-` and future with ‘-b-`.In hindi the past markers are ‘-a`,‘-j-`,‘-i` and future marke is the optative marker ‘-ga`.
For example:

Magahi

i. ʊ sʊt̪- l-o
he sleep-PST-NH
“he slept”

ii. tʊ sʊt̪-b-ə
you sleep-FUT-2P-NH
“you saw”

Hindi

iii. tum -ne dekʰ –a
you-ERG see-PST.M.SG
“you saw”

iv. tu dekʰe-ga
you see-OPT.FUT.M.2SG
“You will see”

(f) In Magahi a plural marker ‘-ən` is added to form plural constructions but this marker is absent in numeral constructions. Whereas in Hindi, plural constructions are formed by adding nasalisation irrespective of any form of constuction. For example:

	Singular	Plural
Magahi	ləikɑ	ləik-ən
	boy	boys
	e-go ləikɑ	du-go ləikɑ
	one-CLF boy	two-CLFboy
Hindi	ləɽka	ləɽkẽ
	boy	boys
	ek ləɽka	do ləɽkẽ
	one boy	two boys

(g) Hindi and Magahi both differ in their lexicon as well.
For example:

Hindi	si:r	ɖʰoop
Magahi	matʰa	rɑuɖa
Translation	head	sunrays.

(h) Adjectives, like nouns, also have two forms in Magahi: a base form and an inflected form. The inflected nouns always take inflected adjectives. Concord between an adjective and a

noun is inflected with number, gender (it should be noted that concord inflecting gender has to be natural sex in case of animates and not the Noun class as it is used in Hindi) and also familiarity (Alok, 2010). Hindi adjectives too show inflection but concord is only with number and gender (both natural and grammatical).
For example:

Magahi

i.	kəri-kɑ	ləik-wɑ
	black-SUF-M	boy-PRT
	"the black boy"	
ii.	kəri-k-i:	ləiki-ɑ
	balck-SUF-F	girl-PRT
	"the black girl"	
iii.	kəri-k-ən	ləik-w-ən
	black-SUF-PL	boy-PRT-PL
	"the black boys"	

Hindi

iv.	kala	ləɽka
	black	boy
	"Black boy"	
v.	kali	ləɽki
	black	girl
	"black girl"	
vi.	kalə	ləɽkə
	black	boys
	"black boys"	

2. Literature Review

This section outlines a brief literature survey of Currently, no tool exists that can identify Magahi from Hindi. One of the reasons for this gap is that Magahi is a less-resourced language. There is a significant lack of computational resources in this language where one can find only a Magahi POS tagger, Magahi monolingual corpus, and Magahi Morph Analyser available (Kumar et al., 2011; Kumar et al., 2012; and Kumar et al., 2016). Several language identification tools have been developed in Indian languages such as (a) In 2008, OCR-based Language Identification tool was developed by Padma and Vijaya which gave 99% accuracy (Padma et al., 2008). (b) In 2014, text-based language identification system were developed for Devanagari script (Indhuja et al., 2014). (c) In 2016, researhers developed a language identifier system for under-resourced languages and it was based on lexicon algorithm which gave an accuracy of 93% (Selamat, 2016). (d) And, in 2017, Patro and others developed language identification tool to disinguish between English and Hindi text based on likelines estimate method with an accuracy of 88%. In this experiment they used social media corpus (Patro et al., 2017).

3. Experimental Set-up

This section isdivided into four sub-sections. It talks about corpus collection and creation, lexicon data-base, extraction of the suffixes, and architecture of the language identifier.

3.1 Data Collection

We have colletcted Magahi and Hindi corpora of 19,884 and 2,00,000 sentences respectively. Magahi data has been taken from the website https://github.com/kmi-linguistics/magahi (Kumar et al., 2016) and Hindi has been crawled from news and blog websites such as Amar Ujala, Live Hindustan, Dainik Jagran, Dainik Bhaskar etc.. We have also used Hindi monolingual corpus from WMT shared task (Bojar et al. 2014) and Indian Language Corpora Initiative (Jha 2010, and Bansal et al. 2013)

3.2 Creation of Lexicon database for Magahi and Hindi

The creation of lexicon database has been prepared using two approaches:

(a)Prepration of unique words:

Magahi		Hindi	
Word	Frequency	Word	Frequency
आउ	5097	इस	19580
ऊ	2857	कर	19141
गेल	2162	कयिा	12839
ओकरा	1443	गया	11602
हलइ	1119	अपने	10361
कहलक	951	रहे	9303
देलक	713	दयिा	8263

Table1: Frequency of Unique words from Magah and Hindi

The unique words for each of the language were extracted using ILDictionary[2], a java-based tool used to create frequency database. The unique words database consisted of 28,548 tokens for Magahi and 1,20,262 tokens for Hindi. In Table1, some words with their frequencies are given.

(b) Extraction of multiple word dictionary:

Magahi	Hindi
हमरा जरूर	पहुँच गया
अड़ध हथुन	ठेका मजदूरों के
कलेजा काढ़ के	बैठकर खाने का

Table 2: Example of Multiple word dictionary

The multiple word groups were prepared upto trie-gram extracted from the corpora. And, for Magahi, we have also included a Morphological Analyser dictionary[3]. Some examples are presented in the table below.

3.3 Extraction of Suffixes

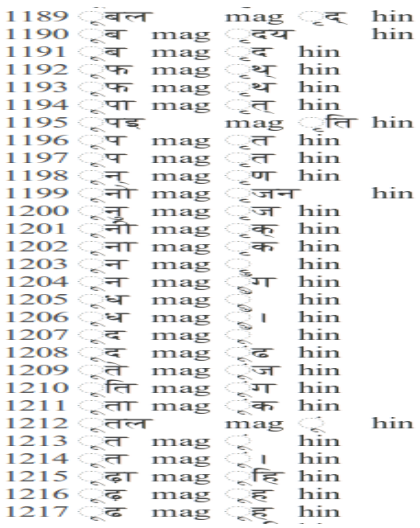

Figure 1: Extracted suffixes of Magahi and Hindi

Suffixes (index) up to 3 characters were extracted from both corpora. Total number of extracted unique suffixes are 8,715 in Magahi and 8,629 in Hindi. Approximately 38.63% of suffixes in these langauges are same such as हित, हलक, ॉलेथा, धक, तैय, डा etc.

3.4 Architecture of Language Identifier

The figure demonstrated below presents the system architecture of the Language Identifier.

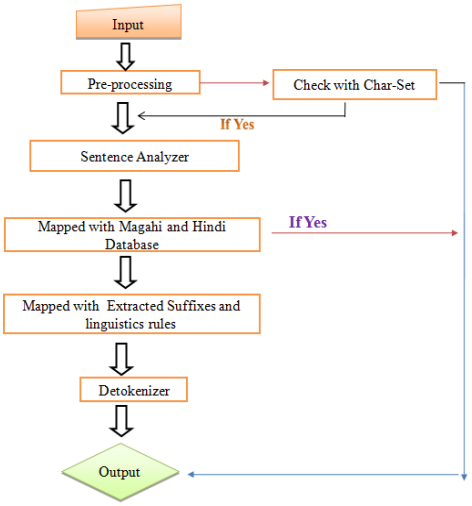

Figure 2: Architecture of Language Identifier

When a user inputs text to the tool, it first goes to the pre-processing section. This section is mapped with the Devanagri char-set. If the input text is in Devanagari then it is sent to sentence analyzer, else it goes directly to the output where tool displays the text belongs from another language. During pre-processing, if some tokens exist in other script then a hidden value is given to those tokens. In the next step, the input text goes to the sentence analyzer where it is tokenized at the word level. After tokenization, it goes for mapping with Magahi and Hindi lexicon data-base simultaneously. If text (tokens or combination of tokens) is mapped with Magahi lexicon data-base then the output"The text is Magahi" is displayed. If the text is mapped with Hindi database then the output"The text is Hindi" is displayed. When the text does/does not matches with both langauges then the system extracts suffix of each word of upto 3 characters. The extraced suffixes are first mapped with Magahi suffixes, through a file containing lingustic rules. If the rule and suffixes do not follow each other then the system cheks Hindi suffixes and its linguistic rules. Thereafter an output is generated in accordance with the mapped lingustic rules. Else an output "Text is of other language" is generated. Before generating

2 http://sanskrit.jnu.ac.in

3www.kmiagra.org/magahi-morph

the final output, the tokens are detokenized. The lingustic rules were prepared on the basis of distinguishing lingustics features of Magahi and Hindi and on the basis of their respective lexicon data-base. The current working system follows the rules on the basis of section 1.2 lingustic features only.

4. Evaluation and Analysis

This system has been evaluated on 2,000 sentences. These sentences came from Hindi, Magahi and other languages. The accuracy of the system has bee evaluated as 86.34%.
The system encountered an error rate of 13.66%. Magahi being a substratum language and Hindi being a superstratum, many lexical items are borrowed in Magahi from Hindi, such as - "फाइल नई दिल्ली ।". The borrwed words create problem the classification of languages. During system analysis, we found other major issues - the system‘s inability to distinguish between the Magahi and Hindi Named Enitities and spelling/typo errors. The system did not prove effective in its ability to tackle short sentences etc. which reduced the system accuracy. Examples of these issues are presented below:

(a) का हो रामौतार ।
(b) तू कौनहैं/हे ।
(c) मात्र पचास रूपइया।
(d) उज्जर बाल ।

(a) type of examples have been identified for both langauges and error takes place due to the presence of named entity.
(b) is a Hindi sentence which has a typo/spelling error which resulted in a structure similar to Magahi.
(c) and (d) type of sentences can appear in both languages. Such short sentences (upto three words) contain words which are common in both languges. However the system identified these as Hindi instead of Magahi.

5. Conclusion

In this paper, we have presented a rule-based language identifier tool to identify a less-resourced language, Magahi, from Hindi. Magahi being closely related to Hindi and a substratum of the, pose greater challanges than unrelated languages.
Future work consists of fixing the above mentioned errors and increasing accuracy of the system. We believe writing heuristics verb anlysis rule can bring significant improvements in the system. We also plan to plug this tool with ILCralwer to improve crawling accuracy. The ILCrawler is used to create the computational framework for collecting Magahi corpus.

6. Acknowledgment

We would like to thank Dr. Ritesh Kumar and his team for providing the Magahi corpora. We would also like to acknowledge the efforts of our colleagues Deepak, Akanksha and reviewers for providing their valubale inputs to improve paper quality.

7. References

Alok, Deepak. (2010). Magahi Noun Particles: A Semantic and Pragmatic Study. Presented at: the “Fourth Students‟ Conference of Linguistics in Indi (SCONLI 4)” at University of Mumbai, 17th-20 th of Feb., 2010.

Alok, Deepak. (2012). A language without Articles: The Case of Magahi. M.Phil.Dissertation, Jawaharlal Nehru University, New Delhi.

Bansal, A., Banerjee, E., & Jha, G. N. (2013). Corpora Creation for Indian Language Technologies–The ILCI Project. In the sixth Proceedings of Language Technology Conference (LTC ‘13).

Bojar, O., Diatka, V., Rychlý, P., Stranák, P., Suchomel, V., Tamchyna, A., & Zeman, D. (2014, May). HindEnCorp-Hindi-English and Hindi-only Corpus for Machine Translation. In LREC (pp. 3550-3555).

Garg, A., Gupa, V., Jindal, M. (2014). A Survey of Language Identification Techniques and Applications. *Journal of Emerging Technologiesnin Web Inteligence.*, 6(4).

Indhuja, K., Indu, M., Sreejith, C., Sreekrishnapuram, P., & Raj, P. R. (2014). Text Based Language Identification System for Indian Languages Following Devanagiri Script. *International Journal of Engineering*, *3*(4).

Jha, G. N. (2010, May). The TDIL Program and the Indian Langauge Corpora Intitiative (ILCI). In LREC.

Kumar, R., Lahiri, B., & Alok, D. (2011). Challenges in Developing LRs for Non-Scheduled Languages: A Case of Magahi. In *Proceedings of the 5th Language and Technology Conference Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC'11)* (pp. 60-64).

Kumar, R., Lahiri, B., & Alok, D. (2012). Developing a POS tagger for Magahi: A Comparative Study. In *Proceedings of the 10th Workshop on Asian Language Resources*, (pp. 10-113), Mumbai, India.

Kumar, R.., Ojha, Atul Kr., Lahiri, B., & Alok, D. (2016). Developing Resources and Tools for some Lesser-known Languages of India. Presented at Regional ICON(regICON) 2016.

Padma, M. C., & Vijaya, P. A. (2008). Language identification of Kannada, Hindi and English text words through visual discriminating features. *International journal of computational intelligence systems*, *1*(2), 116-126.

Patro, J., Samanta, B., Singh, S., Basu, A., Mukherjee, P., Choudhury, M., & Mukherjee, A. (2017). All that is English may be Hindi: Enhancing language identification through automatic ranking of likeliness of word borrowing in social media. *arXiv preprint arXiv:1707.08446*.

Selamat, A., & Akosu, N. (2016). Word-length algorithm for language identification of under-resourced languages. *Journal of King Saud University-Computer and Information Sciences*, *28*(4), 457-469.

Towards a Part-of-Speech Tagger for Awadhi : Corpus and Experiments

Abdul Basit, Ritesh Kumar
Department of Linguistics
Dr. Bhim Rao Ambedkar University, Agra
basitansari03@yahoo.com, riteshkrjnu@gmail.com

Abstract
Awadhi is an Indo-Aryan language, spoken in the eastern region of Uttar Pradesh by approximately 38 million native speakers. However, despite this large number of speakers, it is highly lacking in language resources like corpus, language technology tools, guidelines etc till date. This paper presents the first attempt towards developing an annotated corpora and a POS tagger of the language, The corpus is currently annotated with part-of-speech tags. Since there is no earlier tagset available for Awadhi, the POS tagset for the language was developed as part of this research. The tagset is a subset of the BIS scheme, which is the national standard for the development of POS tagsets for Indian languages.

Keywords: Awadh, POS Annotation, BIS, Corpus development, Less-resourced language

1. Introduction

Awadhi is an Indo-Aryan language, spoken in the eastern region of Uttar Pradesh viz. Lucknow, Raebareli, Sitapur, Unnao, Allahabad, Faizabad, Sultanpur, Behraich and Pratapgarh etc. According to 2001 census, there are 38 millions native speakers of Awadhi language. It is the official language of Nepal and Fiji. Awadhi writing system follows Devanagri, Kaithi and Perso-Arabic script.

In present scenario of India, there are several attempts to collect the corpus of Indian languages and few corpora are available in some of the major languages of India. However, there is no corpus available for Awadhi till now. In the present research, the data of Awadhi language is collected from Eastern region of Uttar Pradesh. In this research, I have developed a corpus with approximately 70,000 tokens. Approximately 20,000 tokens of the corpus data has been annotated with the POS information. It is the first POS-Tagged corpus of Awadhi language. I have also developed the first POS tagset of Awadhi based on the general BIS tagset for Indian language.

The coherent ratio through different varieties of the langage is very rich, but other elements such as affixes, auxiliaries, address terms and domian specific terms differ a lot. The word order of Awadhi is Subject Object Verb (SOV). The use of postposition like मा, से , का etc. indicate possession in Awadhi. Final noun head, two genders ; Musculine and Feminine, clause constituents indicated by case marking. The verbal affixation marks person, number and gender of subject and object. There is an ergative less non-tonal language. There are 30 consonants and 8 vowels phonemes in Awadhi. The writing system fallows Devanagri and Perso-Arabic ascript. The morphological typology of Awadhi language is fusional. (Awadhi/Ethnolgue)

2. Development of POS-Tagged Corpus

In this section, we discuss the process of the development of the post-tagged corpus of Awadhi. It includes the methods of data collection, sources of data, format of data and metadata for current research. It also discusses the challenges and issues in Optical Character Recognition (OCR) of Awadhi texts using a Hindi OCR system and how we worked around the problems. We also discuss the part-of-speech (POS) tagset of Indian languages approved by the Bureau of Indian Standards (BIS) and the POS Annotation tool that we have used for annotating the data.

2.1. Corpus Collection

Corpus provides an empirical base for various linguistic observations, hence, it is a primary source of data for the purpose of linguistic studies and for developing various tools for Computational Linguistics and Natural Language Processing. The ideal aim of data collection is to include as much diversity of a language as possible. As such it is carried out to include millions of words collected from different domains. The current research, however, aims to collect at least hundred thousand tokens of Awadhi language for the corpus formation.

2.2. Source of Data

The data for the current research has been collected from Uttar Pradesh Hindi Sansthan's Library and various publication house in Lucknow. The corpus data has been collected from primary sources i.e., textbooks, short stories and novels. Some of the sources which has been used for data collection include

- Chandawati
- Nadiya Jari Koyla Bhai
- Tulsi Nirkhen Raghuvar Dhama

The current corpus includes data from these novels published in Awadhi.

Lack of Resource – Despite being spoken by a large population, Awadhi lacks electronic as well as other kinds of resources. There is only one website named as *Awadhi kay Arghan* (www.awadhi.org.), where, a very limited number of short stories and poetries are available in Awadhi. Some Facebook pages like *Awadhibhasha, Awadhi Wikipedia etc* claim to promote the cause of Awadhi but they hardly contain writings in Awadhi. There are not any regular electronic newspapers, blogs and magazine available in Awadhi language. The language also does not any published grammar or dictionary available. As such it is a rather challenging task to collect data for the language and even a minimal corpus could prove be very useful.

2.3. Method of corpus creation

In order to expedite the process of creating data, we did not digitise the texts manually. Instead a pipeline of scanning, OCR and proofreading was followed. We expected this process to be much quicker than manually typing out the contents to digitise them. However, this method also had its own set of challenges, which is discussed in the next section.

2.4. Challenges in Optical Character Recognition (OCR) of Awadhi Texts

The data collection process for current research was quite challenging from several perspectives. The very first and basic challenge was the absolute lack of corpus of Awadhi. And so it naturally follows that OCR system is not being developed for Awadhi. As a result, a lot of spelling errors were found in the OCRed Awadhi corpus data when Awadhi texts were scanned using Devanagri OCR system. Some of the most common spelling errors in OCR are mentioned in Table 1.

Spelling Error	Correct words/Matras
ो (तो)	ौ(तौ
े(ले)	ै(लै)
द। द 7	दादा
द् याखौ	द्याखौ

Table 1 : Most common spelling

As we could see, the errors seem to be largely because of the absence of such words in Hindi and the presence of a very closely-related but quite different word in Awadhi it could be hypothesised that these errors might be because of auto-correction by the 'Hindi' OCR system. Such errors necessitated a manual proofreading of the corpus. The proofreading was carried out a Java/JSP-based in-house editing tool 'editit'.

2.5. The Corpus Editing tool: Editit

A corpus editing tool, Editit, was developed using Java/JSP at the backend and runs on Apache Tomcat 8.5 web server. This tool helped in proofreading and correcting the errors that creeped into the corpus data after OCR.

2.6. POS tagsets for Indian Languages

The Penn Tree bank tagset has emerged as for POS tagging of western languages. But Indian languages is much more morphologically rich features. There are a number of POS tagsets designed by several research groups working on Indian Languages. These are, IIT (ILMT) tagset, LDC-IL tagset (Chandra, kumawat & Srivastava, 2014), AUKBC tagset, JNU Sanskrit tagset (JPOS) (Gopal, 2009), MSRI tagset (Baskaran et al., 2008), CIIL- Mysore tagset and BIS tagset (Chaudhary, 2010) is one of them.

Leech and Wilson (1999) espoused the case of standardization of tagset for their reusability of anointed corpora and interoperability across different languages. The result of their effort was EAGLES guidelines. To Achieve the same results BIS has been adopted as the standard for Indian Languages.

BIS is a national-level body that decides on the standard and since this framework (from which the Awadhi tagset was derived) has been approved by BIS, it is now a national standard and is expected to be used by anyone working on POS tagging of Indian languages – and this is the main motivation of using this tagset. Moreover, the BIS framework allows to derive tagsets for different Indian languages; however, the other tagsets are neither accepted as national standards nor are they developed as generic framework, BIS framework was used for building Awadhi tagset.

2.7. Annotation of the data: BIS Tagset

The Bureau of Indian Standards (BIS) Tagset has recommended the use of a common tagset for the part of speech annotation of Indian languages. The tagset, incorporating the advice of the experts and the stakeholders in the area of natural language processing and language technology of Indian languages, has to be followed in the annotation tasks taking place in Indian languages (Chaudhary and Jha, 2011).

Since there is no earlier tagset available for Awadhi, a POS tagset for the language was developed as part of this research. The tagset is a subset of the general BIS tagset. It is used for the POS tagging of Awadhi corpus of approximately 20 thousand tokens. The tagset has 32 different categories including punctuation, residual and unknown category. The complete tagset is given in Table 2.

S.NO.	Categories	Subtypes Level 1	Annotation Convention	Exam-ples
1	**Noun**	**N**	**N**	**मेहरारु ,किताब दारोगा ,मनसेदु**
1.1		Common	N_NN	चश्मा, गिलास, बासन, डाक्टर

1.2		Proper	N_NNP	अब्दुल, योगेश, रीना, अनम
1.3		Nloc	N_NST	ऊपरै, नीचै, आगै, पीछै
2	**Pronoun**	**PR**	**PR**	**वुइ, तुमे, यहे**
2.1		Personal	PR_PRP	वुइ, तुमरे,
2.2		Reflexive	PR_PRF	अपन, हमरे, खुदै
2.3		Relative	PR_PRL	जौ, जिस, जबै
2.4		Recipro-cal	PR_PRC	दुनौ, आपसै
2.5		Wh-word	PR_PRQ	कबहूँ, काहे, का
2.6		Indefinite	PR_PRI	केउ, किस
3	**Demon-strative**		**DM**	**हिंया, हुंआ, जौ**
3.1		Deictic	DM_DMD	हिंया, हुंआ
3.2		Relative	DM_DMR	जे, जौन
3.3		Wh-word	DM_DMQ	के, काहे
3.4		Indefinite	DM_DMI	काउनौ, किस
4	**Verb**		**V**	**गवा, रहन**
4.1		Main	V_VM	कीन, कै, गवा
4.2		Auxiliary	V_VAUX	रहन, रहै, होय, है
5	**Adjective**		**JJ**	**बड़ा, अच्छै**
6	**Adverb**		**RB**	**तेजी,**
7	**Postposi-tion**		**PSP**	**मा, से, का**
8	**Conjunc-tion**		**CC**	**औ, अउर, बल्कि**
8.1		Co-ordi-nator	CC_CCD	औ, बल्कि
8.2		Subordi-nator	CC_CCS	तौ, कि
9	**Particles**		**RP**	**बहुत, हे, ना, भी**
9.1		Default	RP_RPD	भी, ही
9.3		Interjec-tion	RP_INJ	अरै, हे, वाह
9.4		Intensi-fier	RP_INTF	बहुत
9.5		Negation	RP_NEG	नाही, ना, बिना

10	**Quanti-fier**		**QT**	**तनिक, एक, पहिला**
10.1		General	QTF	तनिक, बहुतै, कुछै
10.2		Cardinals	QT_QTC	एक, दुई, छ
10.3		Ordinals	QT_QTO	पहिला , दुसर का, तीसर का
11	**Residuals**		**RD**	
11.1		Foreign word	RD_RDF	**Other than script of the origi-nal text**
11.2		Symbol	RD_SYM	**$,&,*, (,)**
11.3		Punctua-tion	RD_PINC	**., :, ;, “,’, । , ?,!,**
11.4		Unknown	RD_UNK	
11.5		Echo-words	RD_ECH	**खाना-वाना, कुर्सी-उर्सी**

Table 2 : POS Annotation Scheme of Awadhi

As one would notice, BIS tagset bears close resemblance to the LDC-IL tagset. In addition to one type of a category. It also introduces another subtype. BIS tagset groups together unknown, punctuation and residual in one top-level tag – Residual while LDC-IL tagset had three different tags for these. Noun and Pronoun in the two tagsets are almost identical in the two tagsets. Verb (V), too, has the same subtypes – main verb (VM) and auxiliary verb (VAUX). Adjective and Adverb has no subtype whereas we have two new categories in BIS tagset – one is conjunction (CC) which has two subtypes namely coordinator (CCD) and subordinator (CCS). These subtypes were grouped under particle (RP) in LDC-IL tagset. As a result, Particles (RP) in BIS contains Default(RPD), Classifier(CL), Interjection(INJ), Intensifier(INF) and Negation(NEG) as its subtypes. The other category not in BIS tagset is numerals(NUM) – it is replaced by Quantifier(QT), with General(QTF), Cardinal(QTC) and Ordinal(QTO) as its subtypes. Expect for the three categories of adjective, adverb and postposition , all the categories have two or more sub-categories. Moreover, the category of residual, although not part of the language, it is part of the text which is to be annotated and so included in the tagset. See (Appendix)

2.8. Corpus statistics

Overall, the corpus currently consists out 8,532 sentences, amounting to a total of 95,717 tokens. Out of these, 21,256 tokens are currently tagged with part-of-speech information. We are actively working on the development of this corpus and we hope to get 100k pos-tagged token over a period of next few months.

3. Automatic POS Tagger : Experiments and Results

In order to develop an automatic part-of-speech tagger for Awadhi, we have experimented with a tagged corpus of 21,526 tokens that was tagged by a single annotator using the tagset discussed above.

We experimented with 2 classifiers – Decision Trees and Support Vector Machines (SVM) – using the following set of features -

Word-level features : We used the current word, previous 2 words and next 2 words as features

Tag-level features : We used the tags of previous 2 words as features.

Character-level features : We use the first three characters (prefixes) and last three character (suffixes) as features for training

Boolean features : In addition to the above features, we also used the following additional features – *has_hyphen* (1 if the word has hyphen in it), *is_first* / *is_second* (1 if the word is the first / second word in the sentence), *is_last* / *is_second_last* (1 if the word is the last / second last word in the sentence) and *is_numeric* (if the word is a number).

Using these features, the performance of the two classifiers are summed in Table 3 below

Classifier	**Decision tree**	**SVM**
Precision	0.75	0.78
Recall	0.75	0.78
F1	0.75	0.78

Table 3 : Comparison of 2 POS taggers for Awadhi

As is pretty obvious, both the classifiers suffer from a lack of sufficient amount of data. And we expect the results to move closer to the current state-of-the-art in POS taggers as more data comes in.

4. Summing Up

In this paper, we have discussed the creation of a corpus of approximately 95k tokens in Awadhi and the POS-annotation of approximately 26k tokens. We have also discussed the development of an automatic POS tagger for the language which gives a best F1 score of 78 %. The low score could be explained by the minimal amount of data available for training the system – this is expected to improve as more data becomes available. This is a work in progress and over a period of next few months we hope to develop a bigger corpus as well as a better POS tagger.

5. Bibliographical References

AU-KBC tagset. AU-KBC POS tagset for Tamil. Retrieved from http://nrcfosshelpline.in/smedia/images/downloads/Tamil_Tagset-opensource.odt

Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharya, P., Choudhury, M., Jha, G.N., Rajendran, S., Sravanan, K., Sobha, L. and Subbarao K.V.S. (2008).Designing a common POS-Tagset Framework for Indian Languages. In Proceeding of VI workshop on Asian Language Resources, IIIT, Hyderabad.

Chaudhary, Narayan and Girish Nath Jha. (2011). Creating multilingual parallel corpora in Indian Languages. In Proceedings of the 5th Language and Technology Conference : Human Language Technology as a challenge for computer science and linguistics, pages 85 – 89, Poznan, Poland

Leech, Geoffrey and Wilson, Andrew. (1999). [Edited version of Eagles Recommendations for the Morphosyntactic Annotation of corpora. (1996): at http://www.ilc.cnr.it/EAGLES96//annotate/annotate.html.]

Nitish Chandra, Sudhakar Kumawat, Vinayak Srivastava (2014). Various tagsets for indian languages and their performance in part of speech tagging Proceedings of 5 th IRF International Conference, Chennai, 23rd March. 2014 http://www.digitalxplore.org/up_proc/pdf/55-139590032413-17.pdf

Jha, Nath, Girish., Madhav Gopal and Diwakar Mishra. (2009). Annotating Sanskrit Corpus: Adapting IL-POST. Springer Heidelberg Dordrecht, London New york.

IIT-Tagset, A parts-of-Speech tagset for Indian Languages. Retrieved from http://shiva.iiit.ac.in/SPASAL2007/iiit_tagset_guidelines.pdf(30-12-2017)

www.ethnologue.com/language/awa(27-12-2017)

Demo of Sanskrit-Hindi SMT System

Rajneesh Pandey, Atul Kr. Ojha, Girish Nath Jha
Jawaharlal Nehru University
New Delhi, India
{rajneeshp1988, shashwatup9k, girishjha}@gmail.com

Abstract

The demo proposal presents a Phrase based Sanskrit-Hindi (SaHiT) Statistical Machine Translation system. The system has been developed on Moses. 43k sentences of Sanskrit-Hindi parallel corpus and 56k sentences of a monolingual corpus in the target language (Hindi) have been used. This system gives 57 BLEU score.

Keywords: Machine translation, SMT, corpus, evaluation**,** Sanskrit and Hindi, SaHiT

1. Introduction

Sanskrit and Hindi belong to an Indo-Aryan language family. Hindi is considered to be a direct descendant of an early form of Sanskrit, through Sauraseni Prakrit and 1 speaker in India. Today Hindi is widely spoken across the country as well as in some parts of countries like Mauritius etc. According to the Census of 2001[1], India has more than 378,000,000 Hindi speakers.

The knowledge or information source can be accessed by users through translation of the texts from Sanskrit to other languages. Development of a Machine Translation (MT) system like Sanskrit-Hindi (SaHiT) MT can provide faster and easy solution for this task. Therefore, it becomes necessary that the knowledge contained in Sanskrit texts should be translated in Hindi in easy and cost-effective ways. At Present, there are many online MT systems available for Indian languages like Google, Bing Anussaraka, Anglabharati etc. but not for Sanskrit-Hindi. Even lesser work has been done for building SaHiT MT system: (a) "Development of Sanskrit Computational Tools and Sanskrit-Hindi Machine Translation System (SHMT)"[2] project (from April 2008-March 2011), sponsored by Ministry of Information Technology, Government of India, Delhi. It was a consortia project and in the project, 10 institutions/universities were involved[3] and they followed rule-based approach. It is not fully functional and not accessible. And. (b) During the PhD research work, Pandey (2016) has developed SaHiT MT system on Microsoft Translator Hub (MTHub) and Moses platforms. The MT system achieved 35.5 BLEU score on MTHub but it is also not accessible yet. Details study of MT hub training; error analysis and evaluation were reported in Pandey et.al (2016) and Pandey (2016).

Hence, we have demonstrated in this demo only Phrase based Machine Translation (PBSMT) of SaHiT MT system which was trained on Moses.

2. Description of Moses based SaHiT MT System

The first step was the creation of parallel (Sanskrit-Hindi) corpus and monolingual corpus of the target language (Hindi). We prepared 43k sentences. Out of 43k, 25k sentences were collected from the Dept. of Public Relation[4], Madhya Pradesh (MP) government and rests of the data were manually translated. The next step was the collection of a monolingual corpus and 56k sentences were crawled. The detailed statistics are presented below (Table 1):

Sources	**Parallel (sentences)**	**Monolingual (sentences)**
News domain	25 K	49 K
Literature domain (including Panchtantra stories, books & sudharma journal etc.)	18 K	2 K
Health and Tourism domain	0	5 K
The total size of the corpus	43 K	56 K

Table 1 Statistics of Corpus

After collection of data, we conducted several experiments using Moses tool to get good results. The Moses is an open source SMT toolkit which gives permission to automatically train translation model for any language pair i.e. Sanskrit and Hindi Kohen et. al (2007).

For building the system, we followed the processes of tokenization of parallel and monolingual corpus, filtering out long sentences, the creation of language and translation model, tuning, testing, automatic and human evaluation.

Figure 1 demonstrates user interface of the SaHiT MT system. Initially, user gives input text or uploads Sanskrit text file. Once it has entered or uploaded, it

[1] https://www.ethnologue.com/language/hin

[2] http://sanskrit.jnu.ac.in/projects/shmt.jsp

[3] http://sanskrit.jnu.ac.in/projects/SHMT_images/SHMT_PI.pdf

[4] http://mpinfo.org/News/SanskritNews.aspx

goes for preprocessing such as identification of source language and tokenization. When it is finished, input text goes to tuned model where the model file generates target text output. After that translated sentences go for detokenization and that will be displayed on web interface.

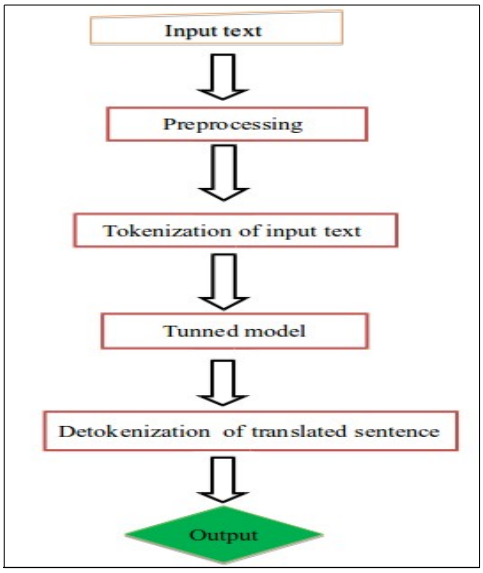

Figure1: Architecture of SaHiT MT System

3. Evaluation

The best MT system was developed after several experiments. Here, we present best of three experiments results of SaHiT MT system.

Experiment phase	Parallel corpus	Monolingual corpus	BLEU score
First	10K	15K	42
Second	26K	40K	54
Third	43K	56K	57

Table 2: Automatic evaluation of SaHiT MT in various phases

In third phase experiments, we have got 57 BLEU score. We have also evaluated on human evaluation parameter to the last phase experiment. This Sanskrit-Hindi MT system was evaluated by three evaluators. They judged the MT output based on the adequacy and fluency. Adequacy and fluency are calculated based on score between 1-5 given by the evaluators. 91% Adequacy and 66.72% Fluency.

The system can be accessed at the following web link: http://sanskrit.jnu.ac.in/index.jsp. Some examples of the MT output are presented below:

i. ते सन्तः वैश्याः सन्ति, ते कदापि न्यूनं न अमान् । (IS[5])
वे सच्चे वैश्य हैं , उन्होंने कभी कम नहीं तोला । (MO)
वे सच्चे वैश्य हैं , उन्होंने कभी कम नहीं तोला । (RT)

ii. मदमुक्तं समाजस्य सकंल्पः नयेयुः। (IS)
नशामुक्त समाज का सकंल्प लें । (MO)
नशामुक्त समाज का सकंल्प लें । (RT)

3.1 Error Analysis -

The MT system encountered several errors. But during the linguistics evaluation, we found the system is not able to produce correct output of target language in the case of Karka relational sentences, Complex sentences, and with Compounding and Sandhi words which reduced the systems accuracy around 68.43 out 100% Pandey 2016). It happens because the system was trained on very small size of corpus. So far this reason, the system is not able to generate. For example:

(a) Issues in Karka level

प्रदेशे महिला सशक्तिकरणाय योजनाः चालयन्ते । (IS)
प्रदेशे में महिला सशक्तिकरणाय योजना चालये जा रहे हैं । (MO)
प्रदेशे में महिला सशक्तिकरण के लिये योजनाएँ चालयी जा रही हैं । (RT)

(b) Issues with complex sentences

महिला–बाल विकास मन्त्रिणी श्रीमती माया सिंहा ग्वालियरे नारी निकेतनतः एकादशमहिलाया पलायिता घटनां गंभीरताया नेयितुम् आयुक्ता मिहला सशिक्तकरणं निरीक्षणं करणस्य निर्देशः दत्तवन्तः सन्ति । (IS)
महिला–बाल विकास मन्त्रिणी श्रीमती माया सिंह ग्वालियर नारी निकेतनत के एकादशमहिलाया भाग गयी घटना का स्मरण गंभीरता से लें आयुक्त महिला सशिक्तकरण निरीक्षण करने के निर्देश दिये गये हैं । (MO)
महिला–बाल विकास मन्त्री श्रीमती माया सिंह ने ग्वालियर में नारी निकेतन से 11 महिला के भाग जाने की घटना को गंभीरता से लेते हुए आयुक्त मिहला सशिक्तकरण को जाँच करने के निर्देश दिये हैं । (RT)

4. Conclusion and Future work

SaHiT attempts to translate Sanskrit text into Hindi language. It gives decent results as compared to previous rule-based MT system or others. Now days it produces 91% adequacy and 66.72% fluency. In future, we will collect more data to train on NMT approach and also work on improving the translation quality of complex and long sentences, and compounding problems etc.

5. References

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., & Dyer, C. (2007, June). Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions (pp. 177-180). Association for Computational Linguistics.

Pandey, Rajneesh. (2016). Sanskrit-Hindi Statistical Machine Translation: Perspectives & Problems. Unpublished PhD. Thesis, Jawaharlal Nehru University, New Delhi.

Pandey, R. K., & Jha, G. N. (2016). Error Analysis of SaHiT-A Statistical Sanskrit-Hindi Translator. Procedia Computer Science, 96, 495-501.

[5] IS= Input Sentence, MO=MT Output
RT= Reference Translation

Demo: Part-of-Speech Tagger for Bhojpuri

Srishti Singh and Girish Nath Jha
Jawaharlal Nehru University,
New Delhi, India
{singhsriss, girishjha} @gmail.com

Abstract

This paper is a demonstration of a POS (Part-of-Speech) annotation tool created for Bhojpuri, a lesser resourced language. Bhojpuri is a popular Indian language and spoken by more than 33 million speakers (census 2001) in India. The digital platform the availability of a good POS tagger is an important requirement for language resource creation and the POS tagger discussed here is one of the initial experiments aiming at language resource creation for Bhojpuri. The tagger was created as part of dissertation work and is based on the BIS (Bureau of Indian Standards) annotation scheme. Tagger performs decently on other varieties of Bhojpuri as well because of the variety of corpus data collected from different sources. The average accuracy achieved by the tool, so far, is 88.6% for general domain.

Key words: Annotation Tool, Bhojpuri POS tagger, Demonstration

1. Introduction

Bhojpuri is language of 33 million speakers majorly in U.P. and Bihar state of India and other countries like Nepal, Bhutan Mauritius, Fiji, Guyana etc. Although, Bhojpuri has gained a lot of attention through Bhojpuri cinema worldwide, it is still struggling for its recognition as a standard language and has no technological resource. Therefore, the motivation behind creating 'Bhojpuri POS tagger' is to bring it to the Digital platform and anticipating other language resource for the language in future. The present POS tagger is one of the pioneering works in this field which is calculated to have an average accuracy of 88.6% which can be found on the following website: (http://sanskrit.jnu.ac.in/bhopos/index.jsp)

2. Bhojpuri POS Tagger

2.1 Tagger description

The general domain representative Bhojpuri Corpus with approx. 192k tokens was created as part a Research work. This is the first big corpus for Bhojpuri. The corpus data is collected from some manually transcribed Bhojpuri folk children stories, websites for literary article, news, magazines, literature etc like bhojpurika.com and anjoria.com with majorly literature, entertainment, politics, sports and blogs etc. The data for corpus creation is collected both manually and semi-automatically using ILCrawler and Sanitizer for collection and corpus cleaning. (Singh, 2015b).

A two-tier hierarchical tagset for Bhojpuri was designed in this endeavour modelled on BIS standards[1] (annotation scheme for all Indian languages). The tagset initially had 33 tags as reported in Singh (2014) but it latter included a new tag label called echo-before (Ech_B) for phrases like 'adalA-badalI[2]' found in both Hindi and Bhojpuri where the second word means *to change* whereas the first word it the echo of the second preceding it (Singh, 2015). The tagger is trained with Support Vector Classificatory model (SVM) for its excelling performance of big data (Giménez, 2004).

2.2 Tagger Architecture

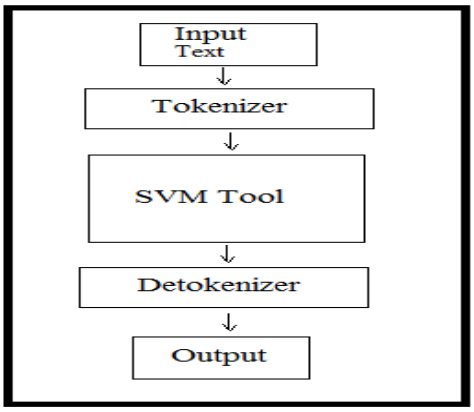

Fig. 1 Architecture of the tagger

Validated raw Bhojpuri corpus is the input for the tool which is first pre-processed and tokenised. The tokenised corpus serves as the input for the SVM machine and the POS tagging is done. The Tagger output s also in tokenised form, therefore, a de-tokenised is used for post-editing is used before displaying the tagged output. The training and test data used for testing is in 80-20 ratio as per the annotation standards.

2.3 Tagger output

Initially the training was performed on a set of 30k tokens and the accuracy of the tagger was calculated to be ranging between 74-85% for random set of data. The latest report on tagger

[1] BIS Guideline: (http://sanskrit.jnu.ac.in/ilciann/index.jsp)

[2] **Itrans** is used for Romanisation the Bhojpuri text throughout the paper.

shows the accuracy of 88.6% when trained on 90k token (Singh, 2015). Currently, the tagger is under development and the training size in being increased along with the size of the Bhojpuri corpus. The Hindi POS tagger trained under ILCI[3] project exhibits an accuracy of approx. 94% at present (Ojha, 2015) which can be found at (sanskrit.jnu.ac.in/pos/index.jsp).

3. Tagger evaluation

Despite belonging to the same language family and sharing much common linguistic features, the use of classifiers, ergative markers, imbedded demonstratives and lexical ambiguity are found in Bhojpuri which are not present in Hindi language. The ambiguity noticed in the corpus is one major challenge for the machine learning. There were ambiguous tokens found in the corpus claiming up to four possible tags for single token as well as four realizations for one tag in different contexts. At the level of POS category, the tagger encountered maximum issues with auxiliary in serial verb constructions; noun & adjectives in conjunct verbs, and ambiguous tokens.

One example of **homophones** cited from Singh (2015b) where 'ka' and 'ke' tokens were often confused with their part of speech category in different contexts. From the corpus it was found to belonging to three possible categories namely subordinator, postposition and auxiliary verb. For example:

1. (kAhe **ke)** sabale manjUra rahale (BHO)
   because all agreed to it (Eng)

2. (lA **ke)** de dA (BHO)
   bring it for him (Eng)

3. hama sUraja DUbe **(ke** bAde) jAiba (BHO)
   I will go only after the sunset (Eng)

The token 'ke' is used as part of *kAhe ke as* subordinator in example 1, *lA ke* as an auxiliary verb in example 2 and *ke bAde* as part of complex postposition in example 3.

Similarly, example of varied **realizations of single token** 'aura' (and) is considered. The conjunction *aura* is represented as '*aura', 'A', 'a',* and '*au'* throughout the corpus as reported in Singh (2014). Moreover, other tool related challenges.

4. Development and Future work

The technological advancement is important for the expansion of a language and resource creation helps retaining and updating the orally transferred knowledge and literature, with time. The present Bhojpuri tagger is an initiative for providing a platform to Bhojpuri and for other NLP tools to come into existence.

The present tagger is under development and both the tagger accuracy and corpus size is being worked upon so that other higher level technological resources can be developed based on the efficiency of the tagger, adding on to the advancement of Bhojpuri.

5. Acknowledgements

We are thankful to ILCI consortium project for providing technical help for the experiment and LREC for considering the paper for demo presentation.

Reference

Choudhary, N. and Jha, G. N. (2011). *Creating Multilingual Parallel Corpora in Indian Languages*. In Proceedings of Fifth Language Technology Conference, Poznan, Poland.

Giménez, J. and Màrquez, L. (2004). *SVMTool: A general POS tagger generator based on Support Vector Machines*. In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal.

Ojha, A., Behera, P. And Singh, S. (2015). *Training & Evaluation of POS Taggers in Indo-Aryan Languages: A Case of Hindi, Odia and Bhojpuri.* In Proceedings of seventh Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Poznań, Poland.

Singh, S and Banerjee, E. (2014). Annotating Bhojpuri Corpus using BIS scheme. In Proceedings of the second Workshop on Indian Languaage Data: Resource and Evaluation (WILDRE). In Proceedings of Ninth International Conference on Language Resource and Evaluation (LREC'14), Reykjavik, Iceland.

Singh, S. (2015a). *Challenges in Automatic POS Tagging of Indian Languages- A Comparative Study of Hindi and Bhojpuri*. Unpublished M.Phil Dissertation submitted to Centre for Linguistics, Jawaharlal Nehru University, New Delhi.

Singh, S. (2015b). *Statistical Tagger for Bhojpuri: Employing Support Vector Machine*. In Proceedings of Forth International Conference on Advances in Computing, Communications and Informatics (ICACCI'15), Kerela, India.

[3] ILCI- **Indian Languages Corpora Initiative** Consortium Project headed by Jawaharlal Nehru University (Choudhary and Jha, 2011)

Developing Resources for a Less-Resourced Language: Braj Bhasha

Mayank Jain[1], Yogesh Dawer[2], Nandini Chauhan[2], Anjali Gupta[2]
[1]Jawaharlal Nehru University, [2]Dr. Bhim Rao Ambedkar University
Delhi, Agra
{jnu.mayank, yogeshdawer, nandinipinki850, anjalisoniyagupta89}@gmail.com

Abstract

This paper gives the overview of the language resources developed for a less-resourced western Indo-Aryan language of India - Braj Bhasha. There is no language resource available for Braj Bhasha. The paper gives the detail of first-ever language resources developed for Braj Bhasha which are text corpus, BIS based POS tagset and annotation, Universal Dependency (UD) based morphological and dependency annotation. UD is a framework for cross-linguistically consistent grammatical annotation and an open community effort with contributors working on over 60 languages. The methodology used to develop corpus, tagset, and annotation can help in creating resources for other less-resourced languages. These resources would provide the opportunity for Braj Bhasha to develop NLP applications and to do research on various areas of linguistics - cognitive linguistics, comparative linguistics, typological and theoretical linguistics.

Keywords: Language Resources, Corpus, Tagset, Universal Dependency, Annotation, Braj Bhasha

1. Introduction

Braj or Braj Bhasha[1] is a Western Indo-Aryan language spoken mainly in the adjoining region spread over Uttar Pradesh and Rajasthan. In present times, when major developments in the field of computational linguistics and natural language processing are playing an integral role in empowering languages, it is essential to include as many languages as possible in this endeavour. It becomes more significant for Indian languages, which are far behind in this area. Following the poor situation of Indian languages in terms of computational resources and applications, there are no available language resources for Braj Bhasha. This is despite the large population of approximately 5 million native speakers. The present work is an attempt to create and develop of some of the basic resources for Braj Bhasha. The paper will focus on the creation and preparation of Text-Corpus, development of BIS based Braj POS Tagset, Universal dependency based annotation of the text corpus at POS level, Morphological features level and Syntactic level.

2. Corpus Creation

This section describes the first-ever annotated text corpus which has been created for Braj Bhasha. Kumar et al. (2016) mention about Braj corpus, however, the details of the corpus is not available.

2.1 Corpus Collection

Though there is a good amount of text written in Braj, these have limitations as almost no data is available in the digital/electronic format. The Braj data has been taken from offline sources which consist of various books and magazines. However, the books and magazines were not easily available. We had to make efforts to collect the text materials. The text we collected mainly belongs to religious texts, short stories, memoirs, culture, art, literary work.

2.2 Digitisation of Corpus

For digitising the available text, an Optical Character Recognition system was developed which makes use of Google OCR. The Google OCR gives a perfect result for Hindi texts written in Devanagari. Since the script of Braj Bhasha is Devanagari, the OCR tool gave quite a satisfactory result for Braj data. The process of digitization is a two-step procedure. First, the individual pages of books and magazines were scanned using a high-resolution scanner. Then, those scanned pages were converted to digital form with the help of the OCR system. At this stage, there were two issues which were needed to be addressed. The first was the cleaning and editing of digitized text because the OCR was meant for Hindi rather than Braj. It required manual editing of the OCRed text. The second issue was the abundance of poetry in some texts, it owes to the fact that Braj Bhasha had been predominantly used for writing poetry. Therefore, the digital data was cleaned to remove poetry text.

2.3 Corpus Statistics

At present, after cleaning, around 5000 pages of a raw unedited corpus is available. As of now, we have edited around 800 pages which consist of around 20,000 sentences and 300,000 tokens. The size of the corpus is being increased on regular basis as more and more data has been edited and cleaned regularly.

3. POS-Tagset and Annotation

POS annotated corpora is a basic resource for several NLP applications. The Braj corpus was annotated using BIS tagset which is shown in section 3.1. In section 3.2, the annotation guidelines for using Braj POS tagset have been discussed. Section 3.3 gives details of POS annotation.

3.1 Braj POS Tagset

There is no POS tagset available for Braj Bhasha as no work has been done on POS annotation of Braj. A Braj POS tagset has been developed for the current research which is based on the BIS[2] guidelines which are a national standard for Indian languages. The BIS tagset has been designed under the Bureau of Indian Standards by the Indian Languages Corpora Initiative (ILCI) group. This tagset takes care of linguistic characteristics of Indian languages. The main characteristic of this tagset is its

1ISO 639-3 Code: bra

2 http://tdil-dc.in/tdildcMain/articles/134692Draft%20POS%20Tag%20standard.pdf

hierarchical nature which takes into account the granularity of linguistic information. The categories at the level 1 are further divided into subtype level 2 and level 3. It is arranged in such a way that the categories at the higher level are more coarse whereas the categories at the lower level are more fine-grained in terms of linguistic/grammatical information. Since We are incorporating morphosyntactic features in UD based morphological and dependency annotation, the hierarchy of Braj POS tagset has been restricted to two levels only.

The Braj POS tagset contains eleven level 1 categories which are divided into 32 fine-grained categories at level 2 of the hierarchy. In the tagset, three level 1 categories: adjective, adverb and postpositions are not divided into sub-categories. Remaining 8 level 1 categories are further divided into sub-categories. The detailed tagset is given in table 1.

Sl. No	**Category**		**Label**	**Annotation Convention**	**Examples**
	Top level	Subtype (level 1)			
1	**Noun**		**N**	**N**	कृष्ण, विसैन,
1.1		Common	NN	N	शब्दन, ग्रंथ
		Proper	NNP	N__NNP	राधा, मथुरा
		Nloc	NST	N__NST	आगै, पीछै
2	**Pronoun**		**PR**	**PR**	मैं, वू, अपनौ
2.1		Personal	PRP	PR__PRP	मेरौ, तू, वू,
2.2		Reflexive	PRF	PR__PRF	अपनौ, अपन
2.3		Relative	PRL	PR__PRL	जो, जिस
2.4		Reciprocal	PRC	PR__PRC	आपस, परस्पर
2.5		Wh-word	PRQ	PR__PRQ	कौन, कित
2.6		Indefinite	PRI	PR_PRI	काऊ, कछू
3	**Demonstrative**		**DM**	**DM**	वू, जे, बिन
3.1		Deictic	DMD	DM__DMD	वे, वू, वा
3.2		Relative	DMR	DM__DMR	जे, बिन
3.3		Wh-word	DMQ	DM__DMQ	कौन,
3.4		Indefinite	DMI	DM__DMI	कछू, कहीं
4	**Verb**		**V**	**V**	है, लिखे, होय
4.1		Main	VM	V__VM	धर, रहतौ,
4.2		Auxiliary	VAUX	V__VAUX	है, हे, रहौ
5	**Adjective**		**JJ**	**JJ**	बड़ौ, सूधी,
6	**Adverb**		**RB**	**RB**	धीरें, जल्दी,
7	**Postposition**		**PSP**	**PSP**	पै, कूँ, कौ, सौं
8	**Conjunction**		**CC**	**CC**	पर, कै,
8.1		Co-ordinator	CCD	CC__CCD	अरु, पर
8.2		Subordinator	CCS	CC__CCS	तौ, कै,
9	**Particles**		**RP**	**RP**	तो, ही
9.1		Default	RPD	RP__RPD	भी, तो, ही
9.2		Interjection	INJ	RP__INJ	अरे, हे, ओ
9.3		Intensifier	INTF	RP__INTF	बहुतई, बेहद
9.4		Negation	NEG	RP__NEG	नाँय, न
10	**Quantifiers**		**QT**	**QT**	एक, थौड़ौ
10.1		General	QTF	QT__QTF	थौड़ौ, बहुत
10.2		Cardinals	QTC	QT__QTC	एक, दो,
10.3		Ordinals	QTO	QT__QTO	पहलौ, दूसरौ
11	**Residuals**		**RD**	**RD**	
11.1		Foreign word	RDF	RD__RDF	A word in a foreign script.
11.2		Symbol	SYM	RD__SYM	For symbols
11.3		Punctuation	PUNC	RD__PUNC	Only for punctuations
11.4		Unknown	UNK	RD__UNK	
11.5		Echo words	ECH	RD__ECH	

Table 1: Braj POS Tagset

3.2 Annotation Guidelines for Braj POS Tagset

The following description is the explanation of POS tags:

3.2.1 Noun (N)

The top-level category of the noun has three sub-categories which are as follows:

3.2.1.1 Common Noun (NN)

Words that belong to the types of common nouns, collective nouns, abstract nouns, countable and non-countable nouns. e.g. शब्दन, ग्रंथा

3.2.1.2 Proper Noun (NNP)

Words that denote the name of a person, place, day etc. e.g. राधा, मथुरा

3.2.1.3 Noun locative (NST)

These words can act as both location nouns and as a part of a complex postposition. e.g. आगै, पीछै

3.2.2. Pronoun (P)

The pronoun is divided into five sub-categories:

3.2.2.1 Personal Pronoun (PR)

These encode person feature in them. e.g. मेरौ, वू

3.2.2.2 Reflexive Pronoun (PRF)

It refers to a noun or pronoun which precedes it. e.g. अपनौ

3.2.2.3 Relative Pronoun (PRL)

It links two clauses in a single complex clause. e.g. जो, जि

3.2.2.4 Reciprocal Pronoun (PRC)
These words show reciprocity. e.g. आपस, परस्प्रा

3.2.2.5 Wh-word (PRQ)
Pronouns which falls into Wh-question category. e.g. कौन, कित

3.2.2.6 Indefinite Pronoun (PRI)
Words refer to something indefinite. e.g. काऊ, कछू

3.2.3. Demonstrative (DM)
Demonstratives have the same form as Pronouns. They are always followed by a noun which they modify. Whereas, a pronoun is used in place of a noun.

3.2.3.1 Deictic (DMD)
These are mainly personal pronouns like बे, बू, वा. But these must occur before a noun.

3.2.3.2 Relative (DMR)
It has the same form as a relative pronoun, but it should occur before the noun it modifies e.g. जे, बिन

3.2.3.3 Wh-word (DMQ)
It is same as Wh-pronoun and it should occur before the noun it modifies e.g. कौन

3.2.3.4 Indefinite (DMI)
It has the same form as an indefinite pronoun. It should occur before the noun it modifies. काऊ, कछू

3.2.4. Verb (V)
The verb is divided into two categories of Main and Auxiliary:

3.2.4.1 Main Verb (VM)
The main verb expresses the main predication of the sentence. It can be in the root form or one of its inflected form. A clause must have a main verb. e.g. धर, रहतौ

3.2.4.2 Auxiliary Verb (VAUX)
An auxiliary verb gives information about inflectional features like tense, aspect e.g. है, हे, रहौ

3.2.5. Adjective (JJ)
The adjectives fall into this category e.g. बड़ौ, लम्बी

3.2.6. Adverb (RB)
Only manner adverbs are annotated using this tag e.g. जल्दी, धीरे

3.2.7. Postposition (PSP)
Postpositions are tagged using this tag. e.g. नै, कौ, कूँ

3.2.8. Conjunction (CC)
A conjunction joins two phrases, clauses, noun, etc. These are divided into two sub-types:

3.2.8.1. Coordinate (CCD)
It joins two or more items of equal syntactic importance. e.g. अरू, पर

3.2.8.2 Subordinate (CCS)
It joins main clause with a dependent clause. It introduces dependent clause e.g. तौ, कै

3.2.9. Particles (RP)
Particles do not decline and they do not fall into any other categories mentioned here. These are divided into four sub-types:

3.2.9.1 Default (RPD)
The default particles are as follows: ही, तो, भी

3.2.9.2 Interjection (INJ)
Words which expresses emotions and gets the attention of people. e.g. अरे, हे, ओ

3.2.9.3 Intensifier (INTF)
Adverbial intensifiers fall under this category. e.g. बहुतई, बेहत

3.2.9.4 Negation (NEG)
The words which indicate negation. e.g. न, नाँय

3.2.10. Quantifiers (QT)
It quantifies the nouns. These are divided into three sub-types:

3.2.10.1 General (QTF)
These quantifiers do not indicate any precise quantity. e.g. थौड़ौ, बहुत

3.2.10.2 Cardinals (QTC)
These are absolute numbers either in digits or numbers e.g. 1, 3, एक, दो

3.2.10.3 Ordinals (QTO)
These include ordered part of the digits. e.g. पहलौ, दूसरौ

3.2.11. Residuals (RD)
These categories are the words which are not an intrinsic part of the language. These are divided into five sub-types:

3.2.11.1 Foreign Words (RDF)
The words are written in a non-Devanagari script. e.g. and

3.2.11.2. Symbol (SYM)
It is used for symbols like $, %. # etc

3.2.11.3 Punctuation (PUNC)
It is used for punctuations e.g. ( ) , " ।'

3.2.11.4. Unknown (UNK)
In this category, those words are kept whose annotation cannot be decided.

3.2.11.5 Echo words (ECH)

It is used for words formed by a morphological process known as Echo-formation. e.g. पानी-वानी

3.3 POS Annotation

For the POS annotation, the syntactic function of the word is given importance rather than its pure lexical category. It is necessary to take syntactic context into consideration so that the appropriate grammatical information of the word can be captured. Therefore, a word may change its lexical category depending on where it occurs in a sentence. Two POS annotated examples are given below:

(1) ब्रजभाषा\NNP कौ\PSP छेत्र\NN आज\NN हू\RPD भौत\INTF ब्यापक\JJ है\VM ।\PUNC

(2) पद्य\NN में\PSP तौ\RPD जे\DMD काम\NN भक्तिकाल\NN में\PSP ही\RP_RPD सम्पन्न\JJ है\V_VM. गयौ\V_VAUX हो\V_VAUX ।\RD_PUNC

3.4 Annotation Tool: WebAnno

An open source, general purpose web-based annotation tool, WebAnno[3] (Eckart de Castilho et al. 2016), is used for the annotation. One of the characteristic features of WebAnno is its suitability for a wide range of linguistic annotations including various layers of POS, morphological, syntactic, and semantic annotations. It allows adding custom annotation layer to facilitate the requirement of the user. It allows the distribution of annotations in various formats. WebAnno also allows multiple annotators to collaborate on a project. It is a flexible, easy-to-use annotation tool.

Figure 1. Annotation Examples in WebAnno

4. [4]Universal Dependency (UD) based Annotation

Nivre, J. et al. (2017) say "Universal Dependencies is a project that seeks to develop cross-linguistically consistent treebank annotation for many languages, with the goal of facilitating multilingual parser development, cross-lingual learning, and parsing research from a language typology perspective."

UD framework addresses several NLP related issues. The problem of varied annotation schemes across languages has been addressed by providing cross-linguistically consistent grammatical annotation. "The annotation scheme is based on (universal) Stanford dependencies, Google universal part-of-speech tags, and the Interset interlingua for morphosyntactic tagsets." (Nivre, J. et al. 2017). The annotation scheme is based on existing standards.

One of the key features of UD framework is inclusivity. It provides universal taxonomy along with the scope to include language-specific extensions.

4.1 Basic Principles of UD

The UD annotation is based on the lexicalist view of syntax, which means that dependency relations hold between words. Words enter into syntactic relations. In lexicalist view, the basic annotation units are syntactic words. Words have morphological features encoded in them. Thus, words are not segmented into morphemes.

4.1.1 Morphological Annotation

The morphological specification of a (syntactic) word in the UD scheme consists of following three levels of representation[5]:

- A lemma representing the semantic content of the word.
- A part-of-speech tag a representing its grammatical class.
- A set of features representing lexical and grammatical properties of the lemma and particular word form.

One of the characteristics of the universal tags and features is that they do not include means to mark fusion words. Fusion words need to be split into syntactic word so that they will get POS tag and feature annotation.

4.1.2. Syntactic Annotation

There are three main ways in which syntactic dependency is marked in the UD framework:

1. The syntactic annotation in the UD scheme marks dependency relations between words.
2. The function words attach to the content words they modify and
3. The punctuation attaches to the head of the phrase or clause.

4.2 Morphological and Syntactic Annotation of Braj Bhasha in UD Framework

Under UD framework, there are over 200 contributors who are working on more than 100 treebanks in over 60 languages around the world. Six Indian languages (Hindi, Marathi, Sanskrit, Tamil, Telugu and Urdu) have been

3https://webanno.github.io/webanno/

4For details, refer to http://universaldependencies.org

5http://universaldependencies.org/u/overview/morphology.html

included in the UD framework. The present work attempts to incorporate the UD framework for annotating the Braj corpus. The following section describes it in detail.

4.2.1. Morphological Features:
Morphological Features are additional lexical and grammatical properties of the word which are not covered by universal POS tags. The format in which a feature is used is Name=Value. A word can have any number of features separated by the vertical bar. For example, Number=Sing|Person=3

The following are some of the morphological features used for Braj Bhasha:

AdpType | AdvType | Animacy | Aspect | Case | Definite | Degree and Polarity | Echo | Foreign | Gender | Gender [psor] | Mood | Number | Number [psor] | NumType | Person | Polite | Poss | PronType | Tense | VerbForm | Voice |

An example of morphological features is given below:

(3) ब्रजभाषा\NNP.Inan.Acc.Fem.Sing कौ\PSP.Gen.Masc.Sing छेत्र\NN.Inan.Acc.Masc.Sing.3 आज\NN.Nom.Masc.Sing.3 हू\RPD भौत\INTF.Deg ब्यापक\JJ है\VM.Ind.Sing.3.Pres.Fin.Act. ।\PUNC

4.2.2. Syntactic dependency:
Syntactic annotation in the UD scheme consists of typed dependency relations between words. The basic dependency representation forms a tree, where exactly one word is the head of the sentence, dependent on a notional ROOT and all other words are dependent on another word in the sentence.
Apart from the basic dependency which is obligatory for all the syntactic annotation, an additional enhanced dependency representation can be incorporated which gives a complete basis of semantic interpretation.
The following are some of the dependency relations used for Braj Bhasha:

acl | advmod | aux | case | cc | ccomp | compound | conj | cop | det | dobj | iobj | mark | mwe | nmod | nsubj | obj | obl | punct | root | xcomp |

An example of dependency relations is given in Table 2:
(4) ब्रजभाषा कौ छेत्र आज हू भौत ब्यापक है ।

S.N.	Token	POS. Morph Features	Dependent on S.N.	Dependency relation
1	ब्रजभाषा	NNP.Inan.Acc.Fem.Sing.3	3	nmod
2	कौ	PSP.Gen.Masc.Sing	1	case
3	छेत्र	NN.Inan.Acc.Masc.Sing.3	7	nsubj
4	आज	NN.Nom.Masc.Sing.3	7	nmod
5	हू	RPD\INTF.Deg	4	dep
6	भौत	INTF.Deg	7	advmod
7	ब्यापक	JJ	0	root
8	है	VM.Ind.Sing.3.Pres.Fin.Act	7	cop
9	।	Punc	7	punct

Table 2: UD based dependency for sentence no. (4)

4.2.3 Annotation of Braj using UD

The morphological features and dependency relations, which are given in the previous section, have been used to annotate Braj corpus. At present, we have completed annotation of about 500 sentences. More data is being annotated on regular basis. Once, we have enough data, we would use machine learning approach to train the system. At present, we don't enough data for training purpose. The idea is to create comprehensive resources of Braj so that modern NLP applications can be developed for it.

5. Conclusion

The present research focuses on developing various resources and tools for Braj which does not have any of such resources. There has been some encouraging progress in this regard, as we have been able to create a first-ever digital corpus for Braj. Although it is a raw corpus, it was quite difficult to collect and create the corpus. Further work is in progress, where the digitised corpus is being annotated at different levels of linguistic annotation – POS, morphological features, and syntactic dependencies. Some important and essential resources - annotation guidelines, tagsets, etc - have been created. We are also experimenting with developing automatic tools by using machine learning approach. We have been making progress and hope to present some reliable results during our presentation. Along with these results, we would also discuss the issues and challenges which were faced during the progress the work.
We hope that our work would contribute towards building essential resources of Braj and our methodology would encourage such work for other less-resourced languages.

6. Acknowledgements

A special thank should go to Dr Ritesh Kumar for providing us with various tools to help us in the development of the corpus. A special acknowledgement should be made for Prof Girish Nath Jha for providing us encouragement and opportunity to work on Braj language. We would also like to acknowledge the open platform provided by UD team which allowed to use UD framework for our work.

7. Bibliographical References

Eckart de Castilho, R., Mújdricza-Maydt, É., Yimam, S.M., Hartmann, S., Gurevych, I., Frank, A. and Biemann, C. (2016). A Web-based Tool for the Integrated Annotation of Semantic and Syntactic

Structures. In Proceedings of the LT4DH workshop at COLING 2016, Osaka, Japan.

Kumar, R., Ojha, A. K., Lahiri, B., and Alok, D. (2016). Developing Resources and Tools for some Lesser-known Languages of India. Presented in Regional ICON(regICON) 2016, IIT-BHU, Varanasi, India.

Ojha, A. K., Behera P., Singh S., and Jha, G.N. (2015). Training & Evaluation of POS Taggers in Indo-Aryan Languages: A Case of Hindi, Odia and Bhojpuri. In the proceedings of 7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics. Pages 524-529. Poznań, Poland.

Nivre, J., Agić, Ž, and Ahrenberg, L. et al. (2017). Universal Dependencies 2.1, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, http://hdl.handle.net/11234/1-2515.

https://www.ethnologue.com/language/bra Accessed on 11-01-2018

Demo: Graphic-based Statistical Machine Translator

Atul Kr. Ojha, Girish Nath Jha
Jawaharlal Nehru University
New Delhi, India
{shashwatup9k, girishjha}@gmail.com

Abstract

In this demo proposal, we present Graphic-based Statistical Machine Translator (GBSMT). This tool has been developed on Moses with the purpose of visualizing GBSMT. Currently, it provides facility to train, test and evaluate statistical machine translation on phrase-based and factor-based approach for any language pair.

Keywords: Statistical Machine translation, Moses, corpus, automatic evaluation

1. Introduction

In the last two decades the field of MT has witnessed a rapid growth. Presently, researchers, developers, users and commercial organization are following Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) approaches to build their MT systems. Among these two, SMT is most popular because due to its ability to produce better results even on a small corpus as compared to NMT. The latter requires longer training time which further depends upon system configuration. For instance, if the system is trained on GPU-based machine then time taken is less than CPU which can take up to three weeks. People also use SMT because of MOSES – an open source SMT toolkit which gives permission to automatically train translation model for any language pair i.e., English and Hindi with different language model tools (Koehn, 2007).

However, a disadvantage it carries is that one needs to memorize several commands and processes to build any SMT system like: tokenization, filter to long sentences, language model, translation model, tuning and decoding etc. Missing out any of the above mentioned process or typing a wrong command, gets us an error or a bad SMT system. Such problems occur because SMT works only by command line.

Through this work, we attempt to reduce these problems. In this system, there is no need to remember commands because the same toolkit is used internally for the process of visualization which is presented briefly in the next section.

As per our knowledge, Tilde MT[1] is the only other graphical and cloud-based SMT training platform which is based on Moses toolkit. It was developed under the Let's MT Project (Vasiljevs et al., 2012). But it is available at a cost (free 30-day trial) and is mainly focused on European languages. Our training platform is primarily for Indian languages and is available for researchers at no cost. We hope to provide an impetus to researchers working on Indian Languages MT systems.

2. Architecture of GBSMT

Figure1 demonstrates the GBSMT structure, a web server-based application.[2] After logging in users should upload parallel (including source and target language) and monolingual (target language) files in '.txt' or '.xls/.xlsx/.ods' format. The remaining processes, thereafter, are automated described below:

(a) Pre-processing: Here, the system identifies source and target language scripts, match sentences of source and target files. Further, tokenization, extraction of tuning and test file from the parallel corpus, true-casing etc take place.

(b) Creation of Language and Translation model: Here the system creates language model and translation model from monolingual and parallel files respectively.

(c) Tuning: The system prepares tuning model through decoder method.

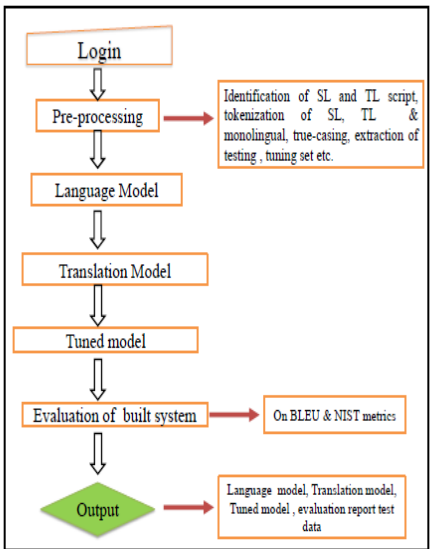

Figure1: Architecture of GBSMT

Once the above processes are over, the system is ready for testing and evaluation. It generates evaluation report from the testing set on BLEU and NIST metrics. Users

[1] https://www.tilde.com/products-and-services/machine-translation/free-trial

[2] http://sanskrit.jnu.ac.in/gbsmt/index.jsp

are then able to download files like Language model, Translation model, Mert file/model, evaluation report etc.

3. Summing up

GBSMT is a tool where users, developers, researchers easily train and build SMT system on window platform. At present people can use this system to build phrase-based statistical machine translation system. But in future, we will include factor-based, automatic evaluate different MT system and other methods to train SMT systems.

4. Reference

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., & Dyer, C. (2007, June). Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions (pp. 177-180). Association for Computational Linguistics.

Vasiļjevs, A., Skadiņš, R., & Tiedemann, J. (2012, July). LetsMT!: a cloud-based platform for do-it-yourself machine translation. In Proceedings of the ACL 2012 System Demonstrations (pp. 43-48). Association for Computational Linguistics.

"Taking Events" in Hindi. A Case Study from the Annotation of Indian Languages in IMAGACT

Massimo Moneglia, Alessandro Panunzi, Lorenzo Gregori
LABLITA – University of Florence
{massimo.moneglia, alessandro.panunzi, lorenzo.gregori}@unifi.it

Abstract

IMAGACT is a cross-linguistic ontology of action, in which action concepts are represented trough Prototypes (3D animations or brief films). The interface IMAGACT4ALL allows mother tongue informants to assign verbs of their language to each prototype and has been used to implement languages belonging to different families. The Ontology specifies the range of different actions which may fall in the extension of each action verb and the set of verbs which can identify each entry, ensuring an adequate translation to action verbs, which show high ambiguity and cross-linguistic semantic variability. A large initiative for the implementation of various Indian languages (Hindi, Urdu, Sanskrit, Bengali, Odia, Assamese, Magahi, Manipuri, Tamil) was undertaken. The paper sketches the status of the work, whose main achievement is the full implementation of Hindi/Urdu and focus on "taking events", that are very relevant in ordinary communication, but feature strong differences in lexical encoding cross-languages. Hindi requires 7 different verbs to cover the actions extended by the general verb take. The main translator लेना(lenA) is also a general verb, but its application has specific semantic boundaries. The paper specifies how features are induced from prototypes, exploiting IMAGACT for the semantic interpretation of Hindi verbs.

Keywords: Action Ontology, Verb Semantics, Comparative Semantics

1. Indian Languages in IMAGACT

IMAGACT is a cross-linguistic ontology of action, in which the entries are prototypic 3D animations (or brief films), each one representing a distinct action concept. Concepts in IMAGACT are connected to a wide set of action verbs with strong impact in the language use: the selected verbs are the ones with highest frequency in speech corpora (Moneglia, 2014; Moneglia and Panunzi, 2007). The ontology of Action (1,010 concepts in the first release) has been induced through a controlled methodology (Moneglia et al., 2012) from English and Italian spoken corpora (Moneglia et al., 2014), grounding relevant concepts on the actual actions referred therein.

The outcome of the induction process leads to specify the set of Italian and English verbs which can be used to refer to each action prototype.

The use of images for action concepts identification allows to extend the «Verb(s)-Action prototype» correlation to any language through competence-based judgments. The web interface IMAGACT4ALL has been designed to allow mother tongue informants to assign verbs of their native language to each entry. Once mapped onto the ontology, each language can be compared to the others. More specifically, the appropriate verb(s) for each action entry (in every implemented language) is specified and the range of action concepts extended by each verb can be compared to the other within and across languages. IMAGACT is therefore a mean to make clear how languages convey a specific semantic categorization of action and also a mean to assist the translation process, specifying what are the verbs required by a given language to identify each particular action type.

IMAGACT have been extended to Chinese and Spanish (Brown et al., 2014) and to a set of languages of different families: Slavonic Languages (Polish and Serbian), Romance languages (Portuguese), German Languages (German and Danish), Arabic and Japanese. Moreover a specific campaign for implementing Indian languages has been undertaken (Moneglia et al., 2014). So far nine languages belonging to three language families has been considered: Sino-Tibetan (Manipuri), Dravidian (Tamil) and Indo-Aryan (Sanskrit, Hindi, Urdu, Odia, Bengali, Assamese, Magahi).

Table 1 specifies the number of processed entries in the Ontology and the number of Action verbs recorded for each Indian language under processing.

Language	Processed scenes	Verbs	Average Scenes per Verb
Assamese	150	103	1.46
Bangla	260	246	1.48
Hindi	1,006	512	2.39
Magahi	100	68	1.59
Manipuri	100	64	1.56
Oryia	110	178	1.28
Sanskrit	212	292	1.83
Tamil	100	95	1.19

Table 1: Number of processed scenes, inserted verbs and the average number of scenes per verb.

Issues and challenges regarding Urdu action verbs have been discussed by Muzaffar et al. (2016). Behera et al. (2016) focused on the possible benefit of this data base for translation. The full implementation of Hindi / Urdu in IMAGACT is a crucial milestone and creates now the possibility of large scale comparison with the other languages and in particular with English. Here we will specifically consider the value of this resource for making objective the peculiar semantic feature which characterizes Hindi verbal lexicon referring to action.

The semantic side is crucial for language disambiguation and translation. There is no one to one correspondence between action concepts and verbs. The number of verbs which can identify one Action may vary from language to language and one verb can in turn identify many different

actions. We call “General” those verbs which share this property.

Verb	Num. Scenes	Verb	Num. Scenes
लगाना (lagAnA)	38	मिलाना (milAnA)	11
रखना (rakhanA)	33	काटना (kATanA)	10
खोलना (kholanA)	30	गिरना (giranA)	10
निकालना (nikAlanA)	24	उतरना (utaranA)	10
उठाना (uThAnA)	21	लाना (lAnA)	9
डालना (DAlanA)	19	पलटना (palaTanA)	9
खींचना (khIMcanA)	18	भरना (bharanA)	9
हटाना (haTAnA)	15	फैलाना (phailAnA)	9
बंद करना (baMda karanA)	14	देना (denA)	9
मारना (mAranA)	13	ले जाना (le jAnA)	9
तोड़ना (to.DanA)	13	छोड़ना (cho.DanA)	8
दबाना (dabAnA)	12	हिलाना (hilAnA)	8
फेंकना (pheMkanA)	12	घुमाना (ghumAnA)	8
बांधना (bAMdhanA)	12	लेना (lenA)	7
गिराना (girAnA)	11	लपेटना (lapeTanA)	7
बंद करना (baMda karanA)	11	खिसकाना (khisakAnA)	7
जोड़ना (jo.DanA)	11	पकड़ना (paka.DanA)	7
चलाना (calAnA)	11		

Table 2: The first 35 general action verbs in Hindi.

Hindi, like English and Italian, characterizes for the presence of many verbs which can be applied to many different Action Concepts (Moneglia et al., 2014). Table 2 specifically presents the Hindi action verbs which can be interpreted according to the larger variety of different prototypes.

This paper is dedicated to the induction of semantic properties of a highly ambiguous language concept, the ones related to «taking events». We will show how a process of semantic feature extraction can be performed starting from IMAGACT. Prototypes and how the procedure should be driven by the annotations which IMAGACT makes available.

2. Taking events in English and Hindi

2.1 The variation of Take across Action Types

One action verb like *to take* is understood by competent speakers as one single action. However, as for many high frequency verbs, it does not refer to a unique action concept, but to many different concepts in the actual language usage. The Figure 1, derived from IMAGACT, shows this phenomenon. The set of prototypes identify how *take* vary its possible reference across different action concepts.

The typological distinction among actions in the extension of one general verb is supported by the fact that different verbs with different meaning are able to identify the same action. Looking to Figure 1, almost each action prototype feature one or more local equivalence with other action verbs, like *to extract*, *to receive*, *to remove*, *to bring*, *to lead*, *to grasp* and so on. This equivalence marks the difference among the represented actions and constitute an explicit differential of each concept prototype with respect to the others.

Once the range of relevant variations and their differential is identified, concepts can be modelled and generalizations obtained. For instance, the set of actions extended by *to take* fall in a restrict set of models roughly

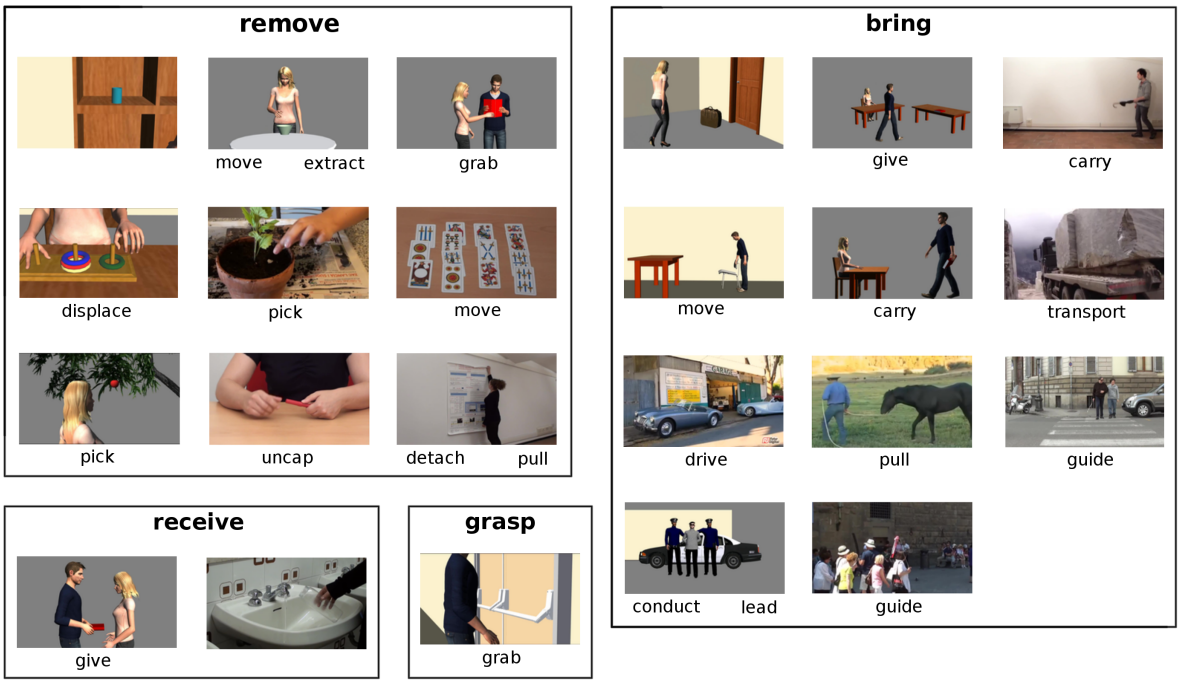

Figure 1: The variation of *to take* across Action Types

identified by a higher level local equivalence (*to remove*, *to receive*, *to bring* and *to grasp*). In conclusion, there are many types of *taking* events which fall under the extension of *to take* and they can be gathered into classes according to high level local equivalence variations, designing the language specific categorization of *taking* events into English.

We do not know exactly what are the boundaries which limit the possible variation of a verbal entry referring to "taking events", however, putting this question at cross-linguistic level, we can see that each language parse the continuum in its own way (Kopecka and Narasimhan, 2012) and starting form IMAGACT data we can make objective what are the differentials among languages.

2.2 Taking events in Hindi

There is not an Hindi verb which covers the full range of applications of *to take*: 7 different verbs are recorded in IMAGACT to satisfy the variation of the English verb, respectively लेना (lenA), पकड़ना (paka.DanA), उठाना (uThAnA), हटाना (haTAnA), निकालना (nikAlanA), लाना (lAnA), ले जाना (le jAnA).

The main translator, लेना (lenA), applies to those taking activities in which the goal is that the "object comes in possession of the agent". This feature can be induced from the small selection of prototypes extended by लेना (lenA) in IMAGACT, compared to the large variation of *to take*. Figure 2 shows a selection of prototypes where both the predicates can be applied face to those that are extended by *take* only (on the right).

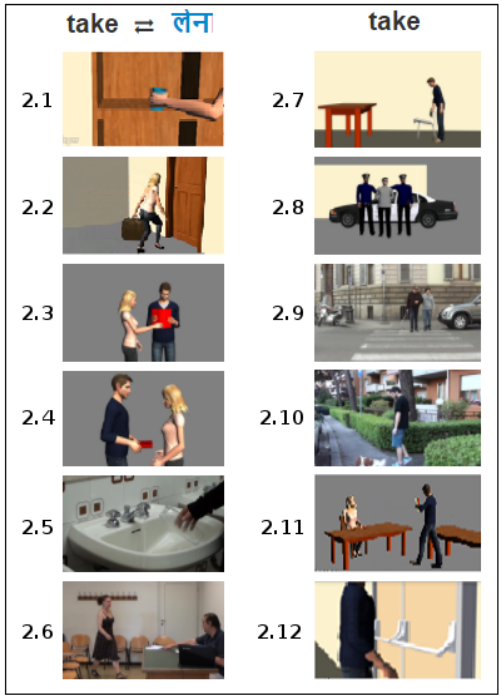

Figure 2: Comparison *take* vs लेना (lenA)

The resulting state "object in possession of the Agent" occurs in all prototypes in which *take* / लेना (lenA) are equivalent, i.e. when "getting object from its location" (2.1), when "getting and bringing the object" (2.2), when "taking is privative of somebody" (2.3), when "taking is also receiving from somebody" (2.4-2.6) or "from a source" (2.5). Under this semantic assumption, it is straightforward the conclusion that the meaning of लेना (lenA) is not appropriate to identify events in which *to take* is equivalent to *to bring* (2.7), *to carry* (2.7), *to lead* (2.8; 2.9; 2.10) and *to give* (2.11), in which the object necessarily have other destinations than the agent.

In parallel, we also find a reason why *grasping* events (2.12) are not extended by लेना (lenA), since the object in these event is "handled", but does not come in the possession of the agent. IMAGACT shows that the verb पकड़ना (paka.DanA) is appropriate in this case (see below).

Those taking events in Figure 1 which the object is extracted from a container or raised from a lower position are respectively captured by the specific predicates उठाना (uThAnA) (to *pick-up / to rise*) and निकालना (nikAlanA) (*to remove / to extract*) (see Figures 3 and 4). However, those events may be also extended by लेना (lenA), which behave as local equivalent of this verb. Indeed, for getting in the possession of an object, we frequently rise or extract it from its collocation[1].

Figure 3: Comparison *take* / उठाना (uThAnA)

IMAGACT makes clear the local nature of the relation of उठाना (uThAnA) and निकालना (nikAlanA) with लेना

[1] It remains unclear from IMAGACT data what are the limits of this equivalence. In some taking prototypes, the object indeed is raised, but उठाना (uThAnA) is not marked in the annotation. The same is when getting an object from a container निकालना (nikAlanA). According to a close evaluation of IMAGACT data for this equivalence relation, the application of लेना (lenA) is possible all the time the object come in the possession of the agent, although the event is categorized by preference with specific predicates. Thanks to Atul Ojha for providing data on this issue.

(lenA). For instance, उठाना (uThAnA) extends to a lot of events where no act of taking is performed. Figures 3 and 4 show the essential of the comparison between the two predicates and taking events. On the right side is displayed the large set of prototypes in which the object is raised or extracted, but no taking event occurs.

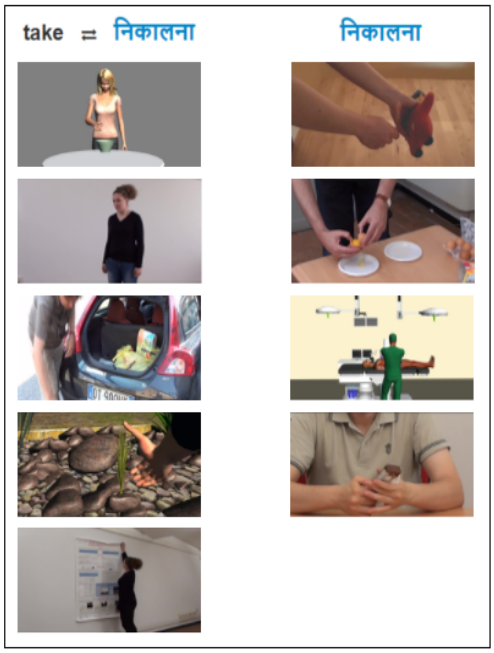

Figure 4: Comparison *take* / निकालना (nikAlanA)

Ragarding taking events where the English verb is equivalent to *remove* and/or *extract*, we can notice that the focus of the taking activity is not the «coming in control of the object», but rather that the object loses its original collocation. This may be the reason why in IMAGACT those prototypes are not marked as the extension of लेना (lenA), which is however marginally acceptable. The appropriate Hindi verbs are respectively निकालना (nikAlanA) (*to extract*) and हटाना (haTAnA) (*take away*).

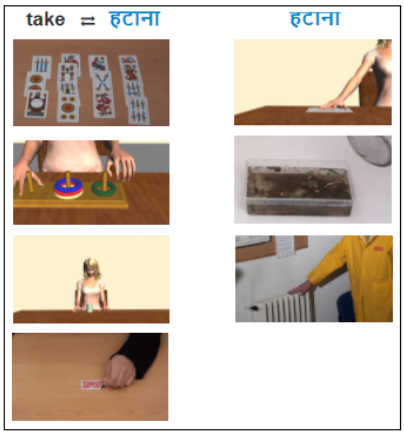

Figure 5: Comparison *take* / हटाना (haTAnA)

Looking at glance to IMAGACT data, the induction of differential semantic features from these prototypes is immediate. Hindi closely distinguish “extractions” from “displacements events”. Indeed, as the comparison in Figure 6 shows, the intersection between हटाना (haTAnA) and निकालना (nikAlanA) in IMAGACT is limited to the events in which displacement is reached through extraction (i.e *extract/remove* a substance from a liquid).

Figure 6: Displacements हटाना (haTAnA) *vs* Extractions निकालना (nikAlanA)

IMAGACT does not gives alternatives to these verbs. It seems that Hindi prefer specific verbs to the general verb लेना (lenA), when removal events take place.
Grasping events are identified by the Hindi verb पकड़ना (paka.DanA), which covers the fields of application where *to take* is equivalent to *to grasp* and *to grab* (Figure 7).

Figure 7: The variation of पकड़ना (paka.DanA)

The range of extensions of the Hindi verb, however, is larger and it over-extends with respect to the range of applications of *to take*, covering also «catching events», that cannot be identified by *take*. The extension of पकड़ना (paka.DanA) to the fields of application of *to catch* is not surprising. For instance the general verbs

coger in Spanish and *prendere* in Italian, can also refer to catching events in local equivalence with other specific verbs (respectively *agarrar* and *acchiappare*).

Contrary to English (and Arabic), bringing events cannot be in the extension of any general Hindi verb referring to the *reaching*, *grasping*, *taking* sequence. Looking to the English variation, in order to predicate of bringing events, the set of equivalent verbs available in the place of *to take*, specifies at least four categories: 1) *bringing / moving* (partially overlapping displacement types); 2) *bringing / giving*; 3) *bringing / carrying*; 4) *bringing / leading*. IMAGACT specifies that Hindi applies two verbs to the events in this variation, respectively लाना (lAnA) and ले जाना (le jAnA).

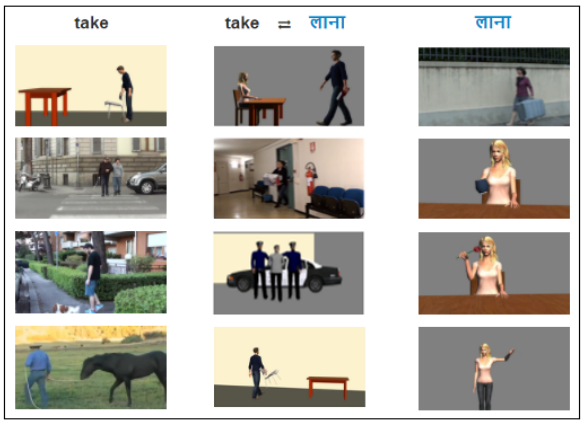

Figure 8: Comparison of *take* / लाना (lAnA)

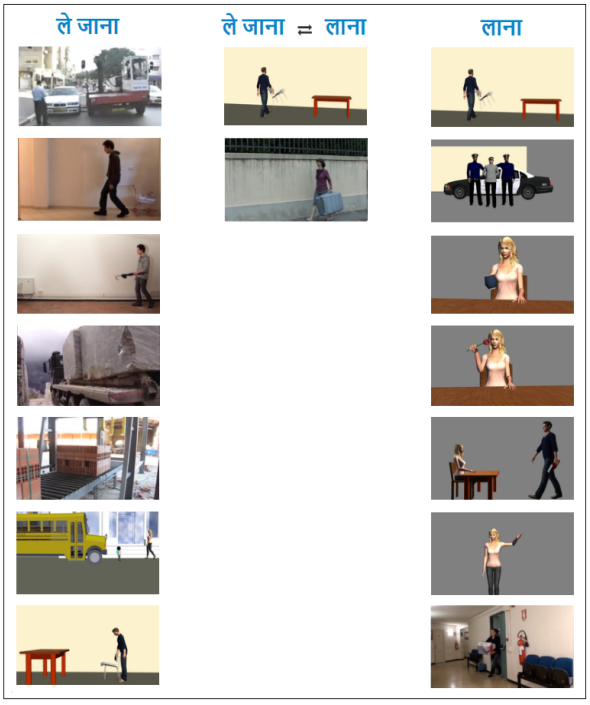

Figure 9: Comparison of ले जाना (le jAnA) *vs* लाना (lAnA)

Figure 8 shows that लाना (lAnA) is quite general, since it can be applied to three categories of bringing events: "bring/give", "bring/move", "bring/carry" (in the centered column), and only shows restrictions on some "bring/lead" events. As Figure 8 also shows, लाना (lAnA) over-extends taking events, since it refers in general to the act of bringing, both when carrying an object in space and when moving an object to a position (right column). ले जाना (le jAnA) partially overlaps लाना (lAnA). As the comparison in Figure 9 shows, both verbs can be applied when movement in space by the subject is accompanied with holding one object (centred column), but ले जाना (le jAnA) appear specifically appropriate to transportation (on the left column) and, contrary to लाना (lAnA), it does not extend to events in which the object is not carried but just moved (right column).

Among the set of action types extended by *take* in Figure 1 IMAGACT does not provide clear results for the identification of leading / guiding events in Hindi, marking with different Hindi verbs similar prototypes, whose differentials are not evident to the user for feature extraction.

Figure 10: Leading events in Hindi

3. Conclusions

The translation of *take* into Hindi and on the other way around the translation in English of the various Hindi verbs that are needed to cover the set of events falling into the *reaching*, *grasping taking* sequence, requires a clear pragmatic knowledge. Verbs are not in translation relation among them, but find their correspondence in specific types of activities. Looking to the set of Action types which falls within the extension of each concerned verb, according to the variation provided by IMAGACT, we figured out that cross-linguistic correspondences follow from semantic regularities. In so doing we have shown that the IMAGACT infrastructure can be used as a core source of information for the semantic modelling of Indian Languages, which, like Hindi, can be compared with other languages on the basis of explicit semantic knowledge.

4. Acknowledgments

The annotation of Hindi has been achieved by prof. Girish Nath Jha who also processed Urdu in collaboration with Sharmin Muzaffar. We also thanks Atul Ojha and the following students for helping in the processing of IMAGACT entries: Himani, Shivek, Shagun, Geeta, Zoya for Hindi; Debmalya for Bangala; Abhijit for Sanskrit; Rajamatangi and Selva for Tamil; Prachi for Marathi; Diksha, Bimrisha and Alvina for Assamese.

5. Bibliographical References

Kopecka, A. and Narasimhan, B. (2012). Events of Putting and Taking, A Cross-linguistic Perspective. Amsterdam: Benjamins.

Moneglia, M., Brown, S. W., Frontini, F., Gagliardi, G., Khan, F., Monachini, M. and Panunzi, A. (2014). The IMAGACT Visual Ontology. An Extendable Multilingual Infrastructure for the Representation of Lexical Encoding of Action. In Nicoletta Calzolari et al., editors, Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 3425-3432. European Language Resources Association (ELRA).

Moneglia M. (2014). Natural Language Ontology of Action: A Gap with Huge Consequences for Natural Language Understanding and Machine Translation. In Vetulani, Z. and Mariani, J., editors, Human Language Technology Challenges for Computer Science and Linguistics, Lecture Notes in Computer Science 2014, 5th Language and Technology Conference (LTC 2011), pages 370-395. Springer.

Moneglia, M., Gagliardi, G., Panunzi, A., Frontini, F., Russo, I., and Monachini, M. (2012). IMAGACT: Deriving an action ontology from spoken corpora. Paper presented at the Eight Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-8). Pisa, October 3-5, 2012.

Moneglia, M., Brown, S. W., Kar, A., Kumar, A., Ojha, A. K., Mello, H., Niharika, Nath Jha, G., Ray, B. and Sharma, A. (2014). Mapping Indian Languages onto the IMAGACT Visual Ontology of Action. In 2nd Workshop on Indian Language Data: Resources and Evaluation (WILDRE-2), pages 51-55.

Muzaffar, S., Behera, P. and Nath Jha, G. (2016). Issues and Challenges in Annotating Urdu Action Verbs on the IMAGACT4ALL Platform. In Nicoletta Calzolari et al., editors, Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pages 1446-1451. European Language Resources Association (ELRA).

Brown, S. W., Gagliardi, G. and Moneglia, M. (2014). IMAGACT4ALL: Mapping Spanish Varieties onto a Corpus-Based Ontology of Action. *CHIMERA* 1:91-135.

Behera, P. Muzaffar, S., Ojha, A. K. and Nath Jha, G. (2016). The IMAGACT4ALL Ontology of Animated Images: Implications for Theoretical and Machine Translation of Action Verbs from English-Indian Languages. In Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP 2016), pages 64-73.

6. Language Resource References

IMAGACT. http://www.imagact.it

Kabithaa: An Annotated Corpus of Odia Poems with Sentiment Polarity Information

Gaurav Mohanty, Pruthwik Mishra, Radhika Mamidi
Language Technologies Research Center
International Institute of Information Technology-Hyderabad
Telengana, India
{gaurav.mohanty,pruthwik.mishra}@research.iiit.ac.in, radhika.mamidi@iiit.ac.in

Abstract

Resource-poor languages, like Odia, inherently lack the necessary resources and tools for the task of sentiment analysis to give promising results. With more user-generated raw data readily available today, it is of prime importance to have annotated corpora from various domains. This paper is a first attempt towards building an annotated corpus of Odia poetry with sentiment labels. Our annotation scheme consists of usage of a polarity identification questionnaire clubbed with taxonomy of emotions. The annotated corpus is further used to build baseline sentiment classification models using machine learning techniques. Stylistic variations and structural differences between poetic and non-poetic texts make the task of sentiment classification challenging for the former. Using the annotated corpus of poems, we obtained comparable accuracy across various classification models. Linear-SVM outperformed other classifiers with an F1-Score of 0.734. The annotated corpus contains a total of 730 Odia Poems of various genres with a vocabulary of more than 23k words. Fleiss Kappa score of 0.83 was obtained which corresponds to near perfect agreement among the annotators.

Keywords: Sentiment analysis, Annotated Corpora, Odia Poems, Machine Learning

1. Introduction

Sentiment analysis entails the extraction and analysis of subjective information present in natural language data. The inception of Web 2.0 served as a gateway to rapid increase in user generated textual content. Opinions are expressed at an ever growing pace in current times on various social media websites. Hence, sentiment analysis systems are widely used for social media monitoring tasks, customer feedback and product review by several commercial organizations [1]. In the area of governance, public feedback via social media and various survey systems is being monitored at a large scale.

With more data available in the native vernacular, the task of sentiment analysis becomes challenging for resource-poor languages. These lack several essential tools and annotated corpora which aid in the task of opinion extraction. Odia is one such language. It is an Indo-Aryan language spoken in various parts of eastern India and has over 45 million native speakers spread across the globe. There is an abundance of Odia data in the form of stories, poems, news articles, blogs, etc over the internet. People have a preference over the genre of textual content they consume depending on their mood. Classification of such content on the basis of the sentiment they evoke, hence is useful. Annotated corpora would therefore be necessary, in order to build automated sentiment classifiers for the same. No such annotated corpus of poems currently exists in literature for any Indian language.

The Kabithaa corpus consists of an annotated collection of 730 Odia poems with sentiment information. Poems and songs are unique among other textual content as they do not follow the same syntactic structure and word order for a language. They contain sentiment information at entity, stanza as well as at document level. These poems have been labelled as either positive or negative. The annotated class is cross-validated by also identifying several emotions that the poems evoke. The baseline experiments have also been conducted in order to compare performance of various classifiers on the annotated corpus.

Kabithaa is the first corpus of Odia poems with annotated sentiment information existing in literature as per our knowledge. It is written in Odia script and hence avoids the pre-processing cost of text normalization. Other than sentiment identification, the corpus is meant to serve as a useful dataset in the task of emotion polarity detection. In terms of application, the emotion tags in the corpus can be used to train models to identify genre of poems. Models can also be trained which help identify user's mood based on the kind of content the user prefers. Identifying the sentiment associated with the poem is the first step towards identifying the emotion(s) which the poem could evoke in the mind of the reader. This can further be used to build recommendation systems which are used by every major company, especially in the e-commerce area.

The paper is divided into 5 sections. Section 2 briefly discusses related work. Section 3 elaborates on creation and annotation of the corpus. The adopted annotation scheme has been provided in the same. Inter-annotator agreement has also been calculated. Section 4 describes the experimental setup for training a model using various classifiers. This helps in establishing the baseline for sentiment classification of these Odia Poems. Possible future work using the annotated corpora is briefly discussed in Section 5.

2. Related Work

So far, sentiment analysis has been majorly focused around classification of non-poetic texts. Moreover, among songs and poems, research on the former is much more than that on the latter. Music classification has been carried out using lyrics (Hu et al., 2009), audio (Lu et al., 2006) and

[1] www.sas.com

even multi-modal features (Laurier et al., 2009) for English. Similar work has been carried out for mood classification of Telugu (Abburi et al., 2016) and Hindi songs (Patra et al., 2016). In the case of traditional literary works such as poetry, a lexicon creation methodology has been discussed for analyzing classical Chinese poetry (Hou and Frank, 2015). The authors propose a weakly supervised approach based on Weighted Personalized Page Rank (WPPR) to create the sentiment lexicon. Such sentiment lexicons are useful for extracting aspect and sentence level sentiment information. Recently a linked WordNet synsets based approach has been proposed in literature to create a sentiment lexicon for Odia (Mohanty et al., 2017). Even though annotated corpora is available for several major languages in India, no such sizable corpus exists for poetic texts, let alone in Odia. Our works aims at bridging this gap.

3. Data Collection and Annotation

User-generated Odia text is readily available over the web. These exist in the form of blog posts, news articles, short stories, poetry, song lyrics, etc. A good amount of these texts however, is present in the form of images of the Odia text, rather than a scrapable text format. Manually converting every single image would be very time consuming. Automatic recognition systems for digitized Odia script documents do exist in literature (Chaudhuri et al., 2001). Fortunately sufficient Odia content is also available in Odia script, in utf-8 character encoding. A few good sources for the same include the Samaja News Website[2] and Odia Wikipedia[3]. For literature, Ame Odia Magazine Website[4] and Odia Gapa[5] serve as useful sources.

3.1. Dataset Source

Odia poems and lyrics are commonly available in transliterated Roman script. For ease of processing, a source where text was available in Odia script was preferred. The **Ame Odia** website hence was the choice for source data. It contains a large and diverse collection of Odia poems along with short stories and blogs. The website has over 800 Odia poems, 400 short stories, 130 essays, and several other texts of various literary forms. At the time of extraction, the website contained 788 Odia poems. These were collected along with meta-data information for each poem. Meta-data includes the title of the poem, name of the poet, and date of publication of the poem.

Since poems differ from prose in syntactic structure and have stylistic variations, pre-processing becomes a necessary step. Instead of sentences, poems are composed of several stanzas and these are sometimes numbered. Stanza numbering does not carry any sentiment information and can be treated similar to functional words. Hence these are removed from the poems. The name of the poet and the date of publication also do not serve the task at hand and therefore are not used in baseline experiments. The title of the poem is retained as it may carry sentiment. **Table 1** provides details on the initial statistics of the dataset before annotation.

Initial Poem Count	788
Total number of Tokens	98782
Total Number of Unique Words	23532
Average Token Count per Poem	∼125

Table 1: Initial Dataset Statistics

[2]http://www.thesamaja.in
[3]http://or.wikipedia.org
[4]http://www.ameodia.com
[5]http://www.odiagapa.com

3.2. Annotation Scheme

A well defined annotation scheme is necessary in order to assign proper sentiment labels to all poems. The task of sentiment analysis can be carried out at three different levels (Liu, 2012). The identification of positive or negative sentiment is carried out at a defined level. Sentiment analysis can be done at an aspect level (Hu and Liu, 2004) or sentence level (or stanza) or at the level of the whole document (Turney, 2002). In the case of poems, it is possible that different parts of the poem elicit different emotions. Since the task is to identify sentiment of the poem as a whole, annotation is carried out only at an overall document level. A polarity identification questionnaire and taxonomy of emotions is provided in order to help the annotators identify sentiment for each poem.

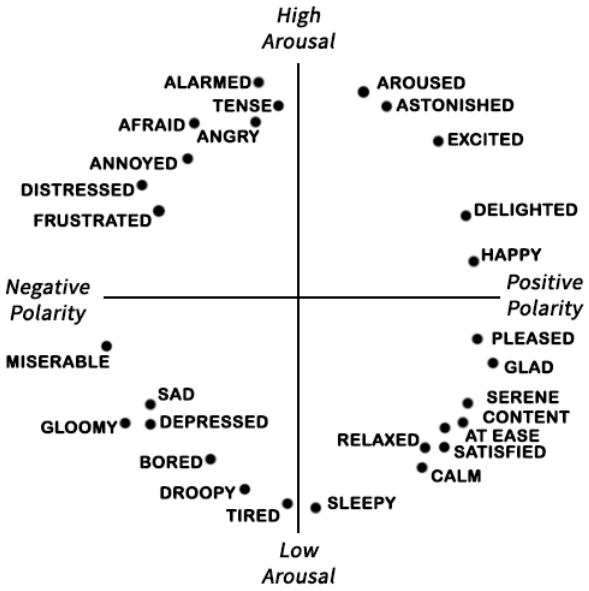

Figure 1: Russell's Circumplex Model classifying 28 Affect words on the basis of positive and negative polarity and arousal.

3.2.1. Identifying emotions

Poems are the most sophisticated form of language (Slinn, 2003). They evoke several emotions in the mind of the reader. In order to detect these emotions, a proper taxonomy is necessary. Russell's Circumplex Model of 28 affect words (Russell and Pratt, 1980) serves as an appropriate reference for emotion identification (Thayer, 1989). The model spots several human emotions on a two dimensional plane of sentiment polarity and arousal as illustrated

in Figure 1. For a given poem, the identified emotions are also tagged by the annotators in order to help validate the poem's annotated sentiment.

3.2.2. Polarity Identification Questionnaire

Once the emotion tags for a given poem are identified by an annotator, the following questionnaire is used to help determine the polarity label for the poem as a whole.
Which of the following best describes **the kind of language the poet is using**?

1. For the whole poem, the poet is using **positive language**, such as expression of support, motivation, admiration, positive attitude, cheerfulness, forgiving nature, positive emotional state, etc. The **emotional states** identified are tending to the **positive side of Russell's model**, for example, happy, excited, calm, serene, etc. [Positive]

2. For the whole poem, the poet is using **negative language**, such as expressions of judgement, negative attitude, sadness, criticism, failure, negative emotional state etc. The **emotional states** identified are tending to the **negative side of Russell's model**, for example, alarmed, angry, miserable, tired, etc. [Negative]

3. The poet is **majorly using positive language** with a minority in the form of negative language. [Positive]

4. The poet is **majorly using negative language** with a minority in the form of positive language. [Negative]

5. The poet is using a **mix of both positive and negative language**, where it is difficult to claim majority of one over the other.

Focus has been given to the language used by the poet. The emotions were identified based on the kind of language used and not by making prior assumptions on what the poet's possible state of mind was when writing the poem. Annotators should not worry about whether they agree or disagree with the poet's views. As the poems are to be classified into two classes, options 3 and 4 focus on determining the sentiment which is expressed more often throughout the poem. The questionnaire along with the identified emotion tags should help determine the dominant sentiment class for each poem. Having option 5 helps annotators in case they are confused about the dominant sentiment for a given poem.

3.3. Evaluation of Dataset

Each Odia poem was annotated as positive or negative by three annotators who are native Odia speakers who speak and write in the language on a daily basis. Poems satisfying options 1 and 3 from the questionnaire were tagged as positive whereas those satisfying options 2 and 4 were tagged as negative. Poems satisfying option 5 were separated from the dataset for future study.
Each annotator was to independently annotate the poems without any communication with the other annotators. The name of the poet was not provided to the annotators as this might induce preconditioned bias. For example, certain poets always write poems which evoke emotions of sadness or anger. Hence the annotator might have a bias towards annotating such poems as negative, given that the annotator knows the name of the poet. A total of 342 poems were tagged as positive whereas 388 poems were tagged as negative. 58 poems were classified as option 5 from the questionnaire. Since the scope of this work entails only positive and negative sentiment classification, these are best kept separated from the final corpus for now. The final classification of a poem was determined by majority rule over all three annotations. Results of annotation are presented in **Table 2**. In order to capture inter-annotator agreement, Fleiss Kappa[6] score for the annotated sample set was also calculated. Inter-Annotator agreement is a measure of how well the annotators make the same annotation decision for the same category. Fleiss Kappa score is calculated with three annotators for two categories (positive/negative) as parameters. A balanced sample set of 342 positive and 342 negative poems is used for Fleiss Kappa (Landis and Koch, 1977). A score of κ **= 0.83** is reported for the Kabithaa corpus which corresponds to "almost perfect agreement".

	Positive	**Negative**	**Total**
Poem Count	342	388	730
Token Count	40546	52142	92688
Removed Poems (Option 5)		58	
Fleiss' Kappa agreement score = 0.83			

Table 2: Results of Annotation

4. Baseline for Sentiment Classification

In order to establish baseline results for the annotated corpus, a few experiments were conducted. The task was to classify Odia poems as carrying positive or negative sentiment by training appropriate classification models. Initially three different classifiers were employed for this task and the results of each were compared. Term frequency-inverse document frequency (TF-IDF) (Sparck Jones, 1972) was used to create a vector representation for an entire poem. We also explore usage of character level n-grams as TF-IDF features and usage of word embeddings to evaluate the performance of these classifiers.

4.1. Experimental Setup

The dataset was split into a ratio of 4:1 for the purpose of training and testing. For initial experiments, TF-IDF features for word n-grams and character n-grams were used for classification. Word and character-based embeddings were also used as features for training the classification models. Experiments were conducted using 'scikit-learn' (Pedregosa et al., 2011), an open source Python library[7]. Precision, Recall and F1-score are the three evaluation metrics which were calculated using **5-fold cross-validation**.
For baseline experiments Naive Bayes, Logistic Regression and Support Vector Machine(Cortes and Vapnik, 1995) were the classifiers used for baseline experiments. LinearSVM was used for training the classification model when using word and character-based embeddings as features.

[6]https://en.wikipedia.org/wiki/Fleiss'˙kappa
[7]http://www.scikit-learn.org

Model	Features	Class	Precision	Recall	F1-Score
Linear-SVM	uni	Negative	**0.682**	**0.707**	**0.691**
		Positive	0.651	0.622	0.632
	uni-bi	Negative	0.675	**0.762**	**0.713**
		Positive	**0.685**	0.582	0.625
	uni-bi-tri	Negative	0.664	**0.796**	**0.727**
		Positive	**0.722**	0.575	0.630
Naive-Bayes	uni	Negative	**0.704**	0.621	0.652
		Positive	0.630	**0.705**	**0.658**
	uni-bi	Negative	0.656	**0.750**	**0.661**
		Positive	**0.675**	0.504	0.523
	uni-bi-tri	Negative	0.640	**0.778**	**0.629**
		Positive	**0.684**	0.396	0.406
Logistic Regression	uni	Negative	0.669	**0.761**	**0.707**
		Positive	**0.682**	0.573	0.615
	uni-bi	Negative	0.626	**0.872**	**0.722**
		Positive	**0.759**	0.406	0.510
	uni-bi-tri	Negative	0.595	**0.924**	**0.718**
		Positive	**0.777**	0.279	0.381

Table 3: Sentiment analysis with Word-Level TF-IDF Features

Model	Features	Class	Precision	Recall	F1-Score
Linear-SVM	(2-6)gram	Negative	0.681	**0.796**	**0.727**
		Positive	**0.722**	0.575	0.630
	(3-6)gram	Negative	0.681	**0.808**	**0.734**
		Positive	**0.728**	0.569	0.631
Naive-Bayes	(2-6)gram	Negative	**0.718**	0.643	0.658
		Positive	0.655	**0.710**	**0.663**
	(3-6)gram	Negative	**0.716**	0.654	**0.660**
		Positive	0.663	**0.700**	0.659
Logistic Regression	(2-6)gram	Negative	0.634	**0.855**	**0.720**
		Positive	**0.756**	0.439	0.534
	(3-6)gram	Negative	0.633	**0.876**	**0.726**
		Positive	**0.781**	0.419	0.520

Table 4: Sentiment analysis with Character-Level TF-IDF Features

4.2. Using TF-IDF Features

TF-IDF assigns weights to words (or n-grams) and serves as a statistical measure for evaluating how important a word is to a document in a corpus. TF-IDF was calculated for unigrams, bigrams and trigrams. **Table 3** illustrates the results of the same for the three aforementioned classifiers.

4.2.1. Using Character n-grams

Even though 730 poems is a sizable corpus for the task at hand, it doesn't show a significant increase in accuracy especially with added bi-gram and tri-gram features. This is because most bi-grams and tri-grams occur sparsely in the entire corpus. In order to tackle the problem of sparsity, we conducted experiments using n-grams at character level. For the baseline, 2-6 and 3-6 character n-grams[8] were used to calculate character level TF-IDF features. The results of the same are illustrated in **Table 4**.

4.2.2. Observation

As observed in Table 3, Linear-SVM performs better with increasing n-grams. It is easy to mistake Logistic Regression to be performing at par with Linear-SVM. However, the former's performance drops drastically for positive class, across the table. Linear-SVM, on the other hand, provides an overall better prediction for both classes.

Through Table 4, it is observed that usage of character level n-grams outperformed that of word-level TF-IDF features for all classifiers. This is because many different words (or n-grams) can share the same character prefixes. Words with common character prefixes should have similar level of importance to a poem. Even when using character level n-grams, Linear-SVM outperformed other classifiers in terms of overall prediction for both positive and negative class. Experimental results show that Precision, Recall and F1-score for poems with negative sentiment are consistently higher than ones with positive sentiment. This could be due to existence of more explicit negative words than positive ones as shown in examples from two different poems in **Table 5**. Several poems manually classified as positive did not have any explicit positive words, yet expressed overall positive sentiment at stanza level and document level. This is because positive sentiment is sometimes not carried through just affect words, but through the overall meaning of the utterance, as shown in the example.

4.3. Word and Character-based Embeddings

In order to obtain useful word vector representations for Odia, we adopted GloVe (Pennington et al., 2014) which

[8]Read as 3 to 6 character n-grams.

Utterance(Roman script)	English Gloss	Theme	Words	Class
Au ethara pachaku dekhani ho sanghaatha daga daga kari chaala, aagaku, aahuri aagaku	Do not look back "Oh" friend, Keep moving forward and forward	Motivating	-	Positive
Anyaaya anithi badhe nithi nithi dayabahi rakhe nighaa	Injustice and dishonesty increases day by day while you keep watching	Complaining about God	Anyaaya Anithi	Negative

Table 5: Examples of explicit positive/negative words in poem stanzas

is an unsupervised learning algorithm for obtaining vector representations for words in any language[9]. Word embeddings are obtained by training on a substantially large corpus of data.

4.3.1. Source

We learned 50-dimensional GloVe word embeddings on Odia news articles corpus [10] containing 500K sentences and 127K unique tokens. Embeddings were computed with GloVe parameters set to default. It is to be noted that only 10K of these tokens overlapped with the Kabithaa corpus vocabulary ($\sim$23K).

4.3.2. Methodology

In order to get vector representation of a poem, the word embeddings for individual words in the poem were used. The mean of all word embeddings for words in a poem was calculated and used as the vector representation for the poem. Linear-SVM was adopted to train the classification model. Since every word in the poem corpus does not have a word embedding, we adopted two different methods to tackle this problem. These are outlined as follows:

1. **Mean of Available Word Embeddings**: Only use available word embeddings for words in a given poem to calculate its mean.

2. **Use Character-based Embeddings**: In order to obtain word embedding for an out-of-vocabulary (OOV) word, we take the mean of the character embeddings of characters which make up the word. A character's embedding was calculated from the GloVe vectors by taking the mean of word embeddings of all the words in the news corpus vocabulary in which that character occurred.

Feauture	Class	Precision	Recall	F1-Score
Word Emb	NEG	0.663	0.763	0.706
	POS	0.682	**0.562**	0.610
Word & Char Emb	NEG	**0.671**	**0.805**	**0.728**
	POS	**0.720**	0.555	**0.619**

Table 6: Results with Word & Character-based Embeddings

[9]https://nlp.stanford.edu/projects/glove/

[10]http://www.thesamaja.com/news_archive.php

4.3.3. Observation

As illustrated in **Table 6**, using character-based embeddings to compute word embeddings for OOV words does show some improvement in performance by the classifier. Comparing this with Table 4 shows that Precision, Recall and F1-Score are comparable to that of using TF-IDF character n-gram features to train the classification models.
The techniques used for baseline experiments leave sufficient room for addition of domain specific features which in turn can provide better performance. Hence the current results may serve as a good baseline for the task of sentiment classification of Odia poems using the Kabithaa corpus.

5. Conclusion and Future Work

Kabithaa is the first corpus of Odia poems, of diverse themes, with poems, manually annotated as either having positive or negative sentiment. In this work, we have described an annotation scheme in which annotators make use of a polarity identification questionnaire along with taxonomy of emotions, for proper assignment of labels to these poems. The emotion tags identified for individual poems are also included in the annotation as meta-data. We have also compared the performance of three classifiers on the Kabithaa corpus. Among all the three, Linear-SVM provides an overall better prediction for both classes. Classification models have been built using both word-level and character-level features. The latter outperforms word level features across all the three classifiers. Using word embeddings as features, it is observed that Linear-SVM gives results comparable to that of TF-IDF character n-gram features. Results of the experiments should serve as a good baseline.
Usage of a Sentiment Lexicon for sentiment extraction at aspect and stanza level can further improve performance of these sentiment classifiers. Stanzas in poetry are usually much longer than sentences in prose. Hence we intend to explore usage of neural network architectures such as BiLSTMs (Graves and Schmidhuber, 2005) in order to capture sentiment at stanza level. BiLSTMs require a much larger training data to work with, hence we intend to focus on sentiment annotation at stanza level. Cue words and Words with sentiment information (lexicons) can also be used to linguistically regularize (Qian et al., 2016) BiLSTMs. We also intend to train classification models on prose data and observe how the same compares with training on poetry data, in the future. Through this paper, we hope the Kabithaa corpus would serve as a good resource in order help evaluate research in sentiment analysis in Odia, especially for sentiment detection in poetic text.

6. Bibliographical References

Abburi, H., Akkireddy, E. S. A., Gangashetti, S., and Mamidi, R. (2016). Multimodal sentiment analysis of telugu songs. In *Proceedings of the 4th Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2016) co-located with 25th International Joint Conference on Artificial Intelligence (IJCAI 2016), New York City, USA, July 10, 2016.*, pages 48–52.

Chaudhuri, B. B., Pal, U., and Mitra, M. (2001). Automatic recognition of printed oriya script. In *6th International Conference on Document Analysis and Recognition (ICDAR 2001), 10-13 September 2001, Seattle, WA, USA*, pages 795–799.

Cortes, C. and Vapnik, V. (1995). Support vector machine. *Machine learning*, 20(3):273–297.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.

Hou, Y. and Frank, A. (2015). Analyzing sentiment in classical chinese poetry. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, LaTeCH@ACL 2015, July 30, 2015, Beijing, China*, pages 15–24.

Hu, M. and Liu, B. (2004). Mining opinion features in customer reviews. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, pages 755–760.

Hu, Y., Chen, X., and Yang, D. (2009). Lyric-based song emotion detection with affective lexicon and fuzzy clustering method. In *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*, pages 123–128.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

Laurier, C., Sordo, M., Serrà, J., and Herrera, P. (2009). Music mood representations from social tags. In *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*, pages 381–386.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Lu, L., Liu, D., and Zhang, H. (2006). Automatic mood detection and tracking of music audio signals. *IEEE Trans. Audio, Speech & Language Processing*, 14(1):5–18.

Mohanty, G., Kannan, A., and Mamidi, R. (2017). Building a sentiwordnet for odia. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 143–148.

Patra, B. G., Das, D., and Bandyopadhyay, S. (2016). Multimodal mood classification framework for hindi songs. *Computación y Sistemas*, 20(3):515–526.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Qian, Q., Huang, M., and Zhu, X. (2016). Linguistically regularized lstms for sentiment classification. *arXiv preprint arXiv:1611.03949*.

Russell, J. A. and Pratt, G. (1980). A description of the affective quality attributed to environments. *Journal of personality and social psychology*, 38(2):311.

Slinn, E. W. (2003). *Victorian Poetry as Cultural Critique: The Politics of Performative Language*. University of Virginia Press.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Thayer, R. E. (1989). The biopsychology of mood and arousal: Oxford university press. *New York*.

Turney, P. D. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, pages 417–424.

Automatic Word-level Identification of Language in Assamese – English – Hindi Code-mixed Data

Manas Jyoti Bora, Ritesh Kumar
Department of Linguistics
Dr. Bhim Rao Ambedkar University, Agra
{manasjyotimj, riteshkrjnu}@gmail.com

Abstract

In this paper, we discuss the automatic identification of language in Assamese – English - Hindi code-mixed data at the word-level. The data for this study was collected from public Facebook Pages and was annotated using a minimal tagset for code-mixed data. Support Vector Machine was trained using the total tagged dataset of approximately 20k tokens. The best performing classifier achieved a state-of-the-art accuracy of over 96%.

Keywords: Code-mixing, Language Identification, Assamese, English, Hindi

1. Introduction

Code-mixing and code-switching in multilingual societies are two of the most well-studied phenomena within the field of sociolinguistics (Gumperz, 1964; Auer, 1984; Myers-Scotton, 1993; Muysken, 2000; Cardenas-Claros and Isharyanti, 2009 and several others). Generally, code-mixing is considered 'intra-sentential' in the sense that it refers to mixing of words, phrases or clauses within the same sentence while code-switching is 'inter-sentential' or even 'inter-clausal' in the sense that one switches to the other language while speaking. In this paper, we will use code-mixing to refer to both these phenomena.

While code-mixing is a very well-studied phenomena within the field of theoretical linguistics, there have been few works computational modelling of code-mixing. In the past of few years, with the explosion of social media and an urgent need to process the social media data, we have seen quite a few efforts at modelling, automatic identification and processing of code-mixing (most notable among them being Solorio and Liu, 2008a; Solorio and Liu, 2008b; Nguyen and Dogruoz, 2013; Das and Gambäck, 2014; Barman, et al., 2014; Vyas et al., 2014 and several others in the two workshops on computational approaches to code-mixing).

In this paper, we discuss the development of an automatic language identification system for Assamese-English-Hindi data at the word level. The data for this purpose was collected from Facebook pages. In the following sections, we discuss some of the previous works, with a focus on Indian languages, the method of corpus collection and annotation and the automatic language identification experiments.

Talking about the languages, English is quite well known and widely used language on online platforms in India. However, Assamese has recently become to be used pretty well in social media by the Assamese people. There was no prior work available on code-mixed social media content in Assamese when we began this particular work. Assamese and Hindi both are Indo-Aryan languages and therefore it is obvious to have many similarities between them, also due to contact and convergence. Specially the lexicons of the two languages have lot of word in common partly finding its root in Sanskrit and due to borrowing. From morphological perspective, we see that they are different in many ways. For instance, Assamese exhibits a rich inflectional morphological but also has agglutinating features in *classifiers* and *case markings*. In Hindi the Phi-features of *person, number* and *gender* are grammatical while in Assamese only the *person* is grammatical. Syntactically both the languages has the basic clause order of SOV.

Annotating the English words did not show much problem per se, however there is a lot of instances of misspelling. But while annotating Assamese and Hindi it was noticed that most of the time the spelling is not in standard form. There are many contractions and usage of non-canonical forms. Besides this there were instances where we saw that a single form was found among the languages which made it difficult to tag its language.

2. Previous Works in the Area

In the past few years, with growing interest and need for processing social media data, there have been quite few attempts at automatically recognising languages in code-mixed scenarios. While language identification at the document level across multiple languages (sufficiently different from each other) is generally considered a solved task, the same could not be claimed about code-mixed data. There have some attempts at language identification in Indian scenario, especially for Hindi-English (Vyas, et al. 2014; Das and Gambäck 2014), Bangla-English (Chanda, et al. 2016; Das and Gambäck 2014) and also Hindi-English-Bangla code-mixed data (Barman et al. 2014). These studies have shown that identifying language at the word-level, especially in the noisy, transliterated data of social media is a very significant and non-trivial task.

Das and Gambäck (2014) is one of the earliest works to address the problem of automatic identification of languages at word-level in social media Hindi-English as well as Bengali-English code-mixed data. They used a flat tagset with 17 tags with separate tags for named entity, abbreviation suffix in a different language. They use a simple dictionary-based method as the baseline and then go on to experiment with SVMs using 4 kinds of features – weighted character n-grams (3 and 4 grams were used), dictionary features (binary feature for each of the 3 languages, decided on the basis of presence / absence of the word in dictionary of a language), minimum edit

distance weight (for out-of-dictionary words) and word context information (3 previous words with tags and 3 following words). The best performing system gave a high precision of over 90% (for Hindi-English texts) and 87% (for Bangla-English texts) but a low recall of 65% and 60% respectively, resulting in an overall F1 score of 76% and 74% respectively for Hindi and Bangla mixed texts. The performance of the system improved by 3% in case of Hindi and 2% in case of Bangla mixed texts

Vyas et al (2014) discuss the development of language identification system for Hindi-English mixed data in the context of developing a part-of-speech annotation system for social media data. They use a different kind of tagset that marks Matrix language of the sentence and Fragment language of the words. They used a word-level logistic regression (King and Abney 2013) for training their language identification system. The system was trained on 3201 English words from a SMS corpus and a separate Hindi corpus of 3218 words. The system gave an overall F1 score of 87% with a very low recall for Hindi data (since the data used for training did not contain spelling contractions and other variations and as such they were labelled as English by the classifier).

Chanda et al (2016), on the other hand, discusses the development of a system for identifying language in Bangla-English texts. They experiment with two different datasets – one from FIRE 2013 and the other of a Facebook Chat which they created. The best performing system makes use of Bangla and English dictionary (and the presence / absence of a word in the dictionary as a binary feature), n-gram and percentage of surrounding words that are predicted as Bangla (again using the dictionary). The model gives an F1 score of 91.5% for the FIRE dataset and 90.5% for the Facebook chat dataset, which is a big improvement over Das and Gambäck's (2014) system but still not quite state-of-the-art.

The state-of-the-art system in identifying languages in code-mixed data in case of Indian languages is discussed by Barman et al (2014). Unlike the other studies, they experiment with a multilingual dataset and train their system on Hindi-English-Bangla code-mixed dataset. They use a tagset with 4 different tags – sentence, fragment, inclusion and wlcm (word-level code-mixing) – each with six attributes. A total of 2,335 posts and 9,813 comments, collected from a Facebook Group, were annotated with these tags. The best performing system was a CRF model trained using 5 different types of features – character n-grams, presence in dictionary, length of words, capitalization and contextual information (previous and next two words) – and it gave an accuracy of 95.76%, closely followed by an SVM model (trained with same features) with an accuracy of 95.52%.

As we could see, all of these approaches make use of language-specific dictionaries to train their models. One of our aims in this paper is to investigate if it is possible to build a reasonably good identification system without the use of a dictionary. Also till now there is no prior work on Assamese-English-Hindi code-mixed data and we plan to make some progress in that direction too.

3. Corpus Collection and Annotation

Since there is no previous corpus available for Assamese-Hindi-English, we collected a large corpus of such data from four different public Facebook pages:

- https://www.facebook.com/AAZGFC.Official
- https://www.facebook.com/Mr.Rajkumar007
- https://www.facebook.com/ZUBEENsOFFICIAL
- https://www.facebook.com/teenagersofassamm

The selection of the Facebook pages was not random. The users in these pages use code-mixing for various reasons. But first of all the users are from different sections of the society. There are different language users dominantly from Assamese who also use English in parallel. Hindi is used by a small number of users, besides Hindi is used in the pages mostly for funny comments with Assamese and also English together. There is one group of people who are seen to code-mix more than others – it is one of the reason for taking the particular pages. This kind of code-mixing has recently become popular among Facebook users. However this is not much common in speaking environments.

The first thing to start with after collecting the data is the annotation. This was done with the tool called 'Webanno' and the code-mixing tagset was used. Heavily inspired by Barman et al. (2014) and Vyas et al. (2014), the annotation scheme has three broad levels of annotation – matrix language (ML), fragment language (FL), and word-level code mixing (WLCM). Each of these levels could be annotated with one of the four language – Assamese (AS), English (EN), Hindi (HI) and Other (OT). The languages other than Assamese (AS) , English (EN) and Hindi (HI) are tagged as other (OT). The punctuations and sybmols are not marked separately and are given the same language name as the word preceding it. As defined earlier in previous works (Myers-Scotton, 1993), matrix language defines the grammatical structure of the sentence while the fragment language refers to the language whose words / phrases are mixed in a clause or a sentence. Word-level code mixing refers to the mixing at the word level – when the base morpheme is of one language and the bound morpheme (especially suffix) is of another language.

The annotation for this was carried out by a single annotator using Webanno. A total of 4768 comments with a total 20,781 tokens were annotated for the task. It took roughly a month to complete the annotation task. The detailed statistics is given in Table 1 below. A list of most frequent Hindi and English words mixed with Assamese (along with frequency of their mixing in the corpus) is also given Table 2.

Languages	**Token Count**
Assamese	11347
English	7689
Hindi	1200

Languages	Token Count
Others	545
Total	**20781**

Table 1: Token Count of each langage in the corpus

English	Frequency	Hindi	Frequnecy
u	147	और	19
you	114	hai	19
the	110	के	13
I	93	में	12
to	79	aap	12
of	76	ka	11
is	73	kya	10
day	73	इंडिया	9
love	62	को	8
a	61	हो	7

Table 2: Most frequent words mixed in Assamese

Let us also take a look at the data and where and why code-mixing occurs in the text.
Comment 1: 'khub enjoy karilu jua kali.'
(Facebook page: Zubeen Garg)

khub enjoy kar-il-u jua kali
much enjoy do-PST-1 yesterday
"I/we enjoyed a lot yesterday (or last night)."

Even though there is a very common word for 'enjoy' in Assamese i.e. 'phurti', still 'enjoy' is used to express the feeling in a more profound way.

Comment 2:
"'জান্না চাহ-তাহ হোঁ মে' [HI] মোৰ কথা হল, গানৰ মাজতো যে আপোনি "চাহ-তাহ" শব্দটো লগাই অসমৰ চাহ পাত [AS] ইন্দাষ্ট্রিৰলৈ (industry) [EN] যি বৰঙণি যোগালে, তাৰ কাৰনে মই অসম চৰকাৰক আপোনাৰ নামত এটা ৰাস্তা নাইবা [AS] এটলিস্ত (at least) [EN] এখন বাইঁৰ দলং, বা এখন কুকুৰা হাইঁ ছাগলিৰ আচনি হাতত লবলে আহ্বান জনাইছো [AS]"
(Facebook page: Mr. Rajkumar)
This comment is a ridicule because of the pronunciation of 'chahta' meaning 'want' as "চাহ-তাহ" which means 'tea~PRD' in a song. By this the commenter says that because the singer used the word 'tea', he has contributed to the Assam tea-industry.

Some of the other examples are given below.
Comment 3:
"Aji Sunday hoi toi gahori bonabi I know"
(Facebook page: Teenagers of Assam)
aji sunday hoi toi gahori khabi I know
today sunday be you pig eat.FUT.2 I know
"Today is sunday so you will eat pork I know."
In this example a complete clause of English is mixed which is very commonly used in conversations among this group of speakers.

Comment 4:
"Moi 4 days continue apunar movie sai world record korim buli bhabisu"
(Facebook page: Teenagers of Assam)

moi 4 days continue apunar movie sai
Gloss: I 4 days continue your movie see.NF

world record korim buli bhabisu
Gloss: world record do.FUT COMP think.ASP.1

"I am thinking that I will make a world record by watching your movies for four days continuously."

4. Experiments

We experimented with Decision Trees and SVM for automatic classification of the language at the word-level (which is the 'fragment language' in our tagset). Our experiments included the following features:
Word Unigrams: This was the most basic feature (and equivalent to the use of dictionary in most of the previous studies).
Word Unigrams and Prefixes and Suffixes (upto 3): Character n-grams have generally proved to be very useful for the task of language identification. Also it has proved to be useful in similar tasks (Berman et al. 2014). Prefixes and suffixes are not actually character n-grams but we expect them to capture similar features of the text. The classifier trained using this feature set formed our baseline classifier.
Contextual Information: Different kinds of contextual information used for the experiments included tag of previous two words and previous and next two words. Again contextual information has proved to be very significant and useful in such tasks.
For the experiments, the data was split into 90:10 ratio with 90% used for training and 10% used for testing.

5. Results

As expected, SVM performed slightly better than the Decision Trees for this task and achieved an average accuracy of 96.01%. A comparative summary of the system's performance with different features is given in Table 3 below.

Features	Classifier	Precision	Recall	F1
Word	SVM	0.78	0.75	0.74
	DT	0.78	0.75	0.74
Word + All prefixes and Suffixes	SVM	0.81	0.81	0.80
	DT	0.80	0.79	0.79
Previous tag		0.93	0.93	0.93
		0.93	0.93	0.93
Previous Tag + Word	SVM	0.95	0.95	0.95
	DT	0.94	0.94	0.94
Previous 2 tags + Word + First Character	SVM	0.95	0.95	0.95
	DT	0.95	0.95	0.95
Previous 2 tags + Previous and next 2 words + word + 3 prefixes and suffixes	SVM	0.96	0.96	0.96
	DT	0.95	0.95	0.95

Table 3: Comparative scores of different feature combinations and classifiers

As could be seen from the above table, tag of the previous word plays probably the most important role in predicting the label of next work. The role of previous tag in such tasks have always been known to be significant and so it was expected that previous tag will play a significant role in the performance of the system. But what was a little surprising is the extent to which it affected the results. In fact, an F1 score of 93% is achieved just by using previous tag as the feature, which is much higher than any other combination of feature. Using words with the previous tags give a further 2% jump. And finally using the prefixes, suffixes and previous and next words, along with previous 2 tags and the word itself leads to a further 1% increase in the performance of the system. As is evident, our approach is language independent and it should work with any other language in a similar way. It pushes the current state-of-the-art by 0.25% and it might be possible to push it further with more data. It must be noted that Hindi is rather underrepresented in the dataset in comparison to English and Assamese. And a preliminary error analysis shows that the classifier achieves a very low precision of 77% with Hindi data. This aspect could definitely be improved with more data. Also further experiments with a sequence labelling algorithm like CRF might improve the results even further.

6. Summing Up

In this paper, we have discussed the creation of the first Assamese-Hindi-English code-mixed corpus, collected from Facebook and manually annotated. This corpus is being made available for further research. We also discussed the development of an automatic language identification system for code-mixed language. Our approach is language independent and could be used for developing similar systems for other languages also. In its current stage, the system gives an accuracy of 96.01%, which is 0.25% higher than the current state-of-the-art. We plan to carry out further experiments and hope to push the performance further up with different algorithms, making changes to the feature set and also by using more data (especially for Hindi) for training.

7. References

Arunavha, C., Das, D. and Mazumdar, C. (2016). Unraveling the English-Bengali Code-Mixing Phenomenon. In Proceedings of the Second Workshop on Computational Approaches to Code Switching, pages 80 – 89.

Auer, P. (1995). The pragmatics of code-switching: A sequential approach. In L. Milroy & P. Muysken (Eds.), *One Speaker, Two Languages: Cross-Disciplinary Perspectives on Code-Switching*. Cambridge: Cambridge University Press, pp. 115-135.

Barman, U., Das, A., Wagner, J. and Foster, J. (2014). Code Mixing: A Challenge for Language Identification in the Language of Social Media. In Proceedings of the First Workshop on Computational Approaches to Code Switching.

Cardenas-Claros, M. and Isharyanti, N. (2009). Code-switching and code-mixing in internet chatting: Between yes, ya, and si a case study. In The jaltcalljournal Vol. 5, No. 3 Pages 67–78.

Danet, B. and Herring, S. (2007). The Multilingual Internet: Language, Culture, and Communication Online. New York: Oxford University Press.

Das, A. and Gambäck, B. (2014). Identifying Languages at the Word Level in Code-Mixed Indian Social Media Text. In Proceedings of the 11th International Conference on Natural Language Processing.

Eliasson, S. (1995). Myers-Scotton Carol, Duelling Languages: Grammatical Structure in Code-Switching. In *Language in Society,* Oxford: Clarendon.

Gumperz, J. John (1964). Hindi-Punjabi Code-Switching in Delhi. In Proceedings of the Ninth International Congress of Linguistics. Mouton: The Hague.

King, B and Abney, S. (2013). Labeling the Languages of Words in Mixed-Language Documents Using Weakly

Supervised Methods. In Proceedings of NAACL-HLT, pages 1110–1119.

Muysken, P. (2000). Bilingual Speech: A Typology of Code-Mixing. Cambridge University Press.

Nguyen, D and Dogruoz, A. S. (2013). Word level Language Identification in Online Multilingual Communication. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 857–862.

Solorio, T. and Liu, Y. (2008a). Learning to Predict Code-Switching Points. In Proceedings of the Empirical Methods in Natural Language Processing

Solorio, T. and Liu, Y. (2008b). Parts-of-Speech Tagging for English-Spanish Code-Switched Text. In Proceedings of the Empirical Methods in Natural Language Processing.

Vyas, Y., Gella, S., Sharma, J., Bali, K. and Choudhury, M. (2014). POS tagging of English-Hindi Code-Mixed Social Media Content. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 974–979, Doha, Qatar, October. Association for Computational Linguistics.

Issues in Conversational Sanskrit to Bhojpuri MT

Shagun Sinha, Girish Nath Jha
Jawaharlal Nehru University, New Delhi
{shagunsinha5, girishjha} @gmail.com

Abstract

The authors of this paper have presented an alpha version of MT system for conversational Sanskrit to Bhojpuri. The paper discusses the challenges in corpora creation for less resourced languages of India, training and evaluation of the MT system in the domain of everyday conversation and the research questions emerging from there.

Keywords: Statistical Machine Translation, Less Resourced Languages Technology, Sanskrit to Bhojpuri.

1.Introduction

With over 30 million speakers as per the 2001 census data, Bhojpuri (ISO : 630-9.[1]) is an Indian language spoken primarily in the state of Bihar in the districts of Champaran, Saran, Shahabad districts; Assam, parts of Uttar Pradesh and West Bengal. [2] It is not listed in the eighth schedule of the Indian Constitution. Bhojpuri is a "morphologically rich and non-configurational language, unlike English" (Behera et al, 2016).

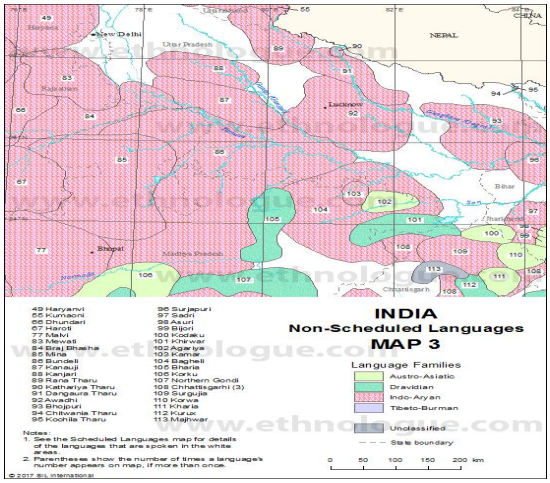

Fig 1. Map of the Non-scheduled languages of India Source: https://www.ethnologue.com/map/IN_03

Bhojpuri is a less resourced language and has found very little space in the technological space. Singh (2014) prepared a POS tagger for Bhojpuri as per the BIS scheme which is the first of its kind work (Singh and Banerjee, 2014). Singh and Jha (2015) proposed a statistical tagger for Bhojpuri based on Support Vector Machine (SVM) whose results were 87.67% for test and 92.79% for gold corpus. A paper based on 'Bhojpuri Annotation' was presented by Neha Mourya in 2015.[3] She also authored papers on "Complex Predicate in Bhojpuri," "Classification of Bhojpuri Adverbials," "Agreement in Bhojpuri"(ibid). Behera et al (2016) worked on "Dealing with Linguistic Divergence in English-Bhojpuri Machine Translation" in which they use Dorr's theoretical framework for resolving divergences between the two languages. Linguistic Divergence "...refers to the concept of structural or 'parametric variation' between a source language (SL) and a target (TL) pair in Machine Translation(MT). In other words, it emerges when the decoded output content lacks 'well-formedness' because of the inherent` linguistic constraints" (Behera et al 1, last accessed Jan 14, 2018). Some research in area of MT is currently in progress at the Jawaharlal Nehru University, New Delhi (JNU) the results for which are currently awaited.

Sanskrit, on the other hand, has been a language of the sub-continent for approximately more than 6000 years. The literary traditions so developed in this 'donor' language are a prized heritage. Many Indian languages have drawn from it from time to time to strengthen their literature. Some MT tools for Sanskrit have been developed with varying results. Pandey (2016) has developed a Sanskrit to Hindi MT as a part of his Ph.D. work at JNU.

In developing resources for Bhojpuri, Sanskrit would prove to be a good literary resource. Additionally, Sanskrit can be taught to Bhojpuri speakers too. To teach Sanskrit to Bhojpuri community, e-learning or other learning technologies can be important. A conversational text would assist in preparing tools for teaching Sanskrit to Bhojpuri speakers.

The current work is a first of its kind at initiating an MT for Sanskrit to Bhojpuri based on a corpora of everyday conversational text.

[1] Ethnologue: https://www.ethnologue.com/language/bho

[2] ibid

[3] https://fmsbhu.academia.edu/NehaMaurya/CurriculumVitae accessed 14th Jan 2018

2. Language Technology Resource (LTR) Creation for Bhojpuri and Linguistic Analysis

The entire set of parallel data was raw and had not been annotated for its properties. The process of parallel corpora preparation had two levels. First, collection of data, and second, translation to Bhojpuri. At the first stage, there were two further possibilities. One, the data had to be directly taken from Sanskrit sources. Second, in cases where the source was not Sanskrit but English (as in case of some news channels), the data had to be first transcribed, then be first translated into Sanskrit. At the second level, the data was finally translated into Bhojpuri for creating the parallel corpora.

2.1. Role of conversational text:

A set of conversational parallel data was one of the key sources of good translations. The parallel corpora contained short sentences of up to 5-6 words. The motive was to train an MT system with corpora that had the least of ambiguities and thus, short conversational texts were preferred. Simple conversational text was taken from news channels to avoid literary or metaphorical usage.

2.2. Corpora Creation

The corpus was created using a multitude of sources of spoken Sanskrit. The online version of Vyavaharsahastri[4] published by Samskrita Bharati (ibid) was used. Online lectures of spoken Sanskrit were transcribed to be translated later (CecUgc)[5]. Not more than 5 YouTube videos of news channels were also used to arrive at a proper conversational Sanskrit text.

The collection of sentences was based on conversational properties- the sentences had simple syntactic structures. The files so created involved naming along the ISO name codes, ie, sa for Sanskrit and bho for Bhojpuri.

In a previous research carried out on a smaller data set, the BLEU was 33.51. For advancing the system performance, the corpora size was increased to a total of 10k sentences. With not more than 7 words each, the first 5k sentences were the simplest set of sentences. The next set of 5k sentences were more complex with 8 words each on an average. After building that system, this set was further divided into different sentence sets for training different systems on even smaller data size. Next, all such separate files of smaller data size were collectively set for training taking the total number of extracted sentence count to 64,843; the number of aligned sentences to 29,669 and the number of used sentences to 59,365. The system thus trained on a corpora which had repetitions of sentences due to repetition of sets. The sentences so collected in the corpora were all utilised for training the system.

Tiwari (1960) has mentioned a detailed analysis of the language spoken in various places. Three types of Bhojpuri are spoken[6], namely, Standard Bhojpuri; Western Bhojpuri and Nagpuria. Standard Bhojpuri is spoken in THE area in and around Bhojpur (ibid).
The standard form has been explained by Tiwari in his work titled 'The Origin and Development of Bhojpuri' (1960). The forms of the male singular words used in Bhojpuri are as given below and have used standards as mentioned by Tiwari and Upadhyay (2008). The post-positions cited by Tiwari are as follows:

Case	Sa	bho
Nominative	Rama	rAma/Ramuvaa[7]
Objective	rAmaM	rAma/ ramuvA'ke' (Tiwari 111)
Instru.	rAmeNa	rAma/ramuvA se (109)
Dative	rAmAya	rAma/ramuvAKAtira
Ablative	rAmAt	rAma/rAmuvA'se' (109)
Genitive	rAmasya	rAmuvA'ke' (111)
Locative	rAme	rAmuvapa/para/ 'mein' (111)

Table1. Post-Positions in Bhojpuri (Tiwari 1960)

Pronouns are similar to standard Hindi except in second person when the use for second person singular is mostly 'tU/tohanI'[8] or rauwA for respect (ibid). For third person, the use goes to 'hama' mostly (ibid).

Verbs formed in Bhojpuri indicate usually the time and person (first, second, third) doing the act also. An example of Present Perfect as cited in Tiwari (1960) 182 is being given here:

Verb	Sanskrit	Meaning
U gayilasa	sah agachchat	He went
U gayalI	sA agachchat	She went
U gayilan	sah agachchat	He went (sense of honor)

Table 2. Verbs in Bhojpuri (Tiwari, 1960 182)

Similar has been indicated for Past perfect which is indicated using the word 'raha' (Tiwari 1960 p185),

[4] Sanskrit Ebook Website, See Ref List

[5] Learn Sanskrit Be Modern series

[6] Upadhyay 21

[7] Upadhyay, 26

[8] Upadhaya 26

Past continuous indicated by 'raha' (Tiwari 1960 p182).

3. Using MT Hub for Training

Microsoft Translator Hub is an extension of Automatic Microsoft Translator API service which has been "designed for organizations that have specific translation needs" (https://www.youtube.com/watch?v=b5qBSlKwDeg) and has been quite useful in training MT systems for research and development. The successful systems can also be deployed for wider testing evaluation by the community.

For the training part, the aforementioned corpora and parallel corpora is utilised. The parallel corpora is built by the Hub upon the upload of each of the Sanskrit monolingual corpora file and the translated Bhojpuri corpora file separately. The two files are merged to form a Parallel corpora file. Additionally, the monolingual Bhojpuri file is uploaded separately for providing monolingual reference.

The training takes place on a set of training data which it takes from the uploaded files. A section of the parallel corpora which is different from the training data is set apart for the parallel file deriving references from the monolingual or target language corpora. Upon training the system on a set of data, the hub tests it and that produces the BLEU score based on the degree of similarity with the Human reference provided in the corpora.[9]

In a previous attempt, the system had a simple set of 5k sentences for parallel data and the BLEU was 33.51. The data size, however, was low and thus, not sufficient.

4. Evaluation of the MT output

For the current research. the result obtained at the end of training was a BLEU score of 37.28. The system took 1 hour 28 minutes to train.

The sentence translations obtained in a total of 42 pages can be divided into these categories:

- Sanskrit words mixed with Bhojpuri & mixed inflections
- Exact Sanskrit Phrase used in Bhojpuri
- Clear Perfect Translations
- Clear meaning despite unmatched references
- Translation better than the reference

4.1. Mixed Translations

These translations included Sanskrit words also. For example, in the following sentences with the Reference pattern given as on the Microsoft Translation platform where P stands for Page, S for Sentence number:

Reference	Sanskrit	Bhojpuri
P5, S2	eSha auShadhih	I **auShadhih** bATe
P3, S1	sImnah ullaMghanaM karoti sarvadA	sImA ke **ullaMghanaM** kare lA hamesA

Table3. Mixed Translations as obtained

These occurrences were prevalent and prominent throughout the results.

Mixed Inflections

Reference	Sanskrit	Bhojpuri
P6, S9	kaH rakShayiShyati deSam	ke **rakShayi Shyati log**

Table4. Mixed Inflections as obtained

In total, the mixed translations occurred in 2-5 out of 10 sentences.

4.2. Exact Sanskrit words:

Reference (as in the Translation platform)	Sanskrit	Bhojpuri
P24, S10	gRhavyavasthA	**gRhavyavas thA**

Table 5. Exact Sanskrit Words

Such translation happened **only in single word** sentences which did not have repeated occurrences.

4.3. Perfect Translations

Reference (as in the Translation platform)	Sanskrit	Bhojpuri
P29, S1	kasmAt sarve bibhyati	kekarA se saba Dare lana
P29, S3	kim abhavat	kA bhayila

Table 6. Perfect Translations as obtained in results

Perfect translations ranged from 0-3 occurrences for every 10 sentences.

[9] https://www.youtube.com/watch?v=-UqDljMymMg last accessed 03.03.2018 at 23.50 hrs IST

4.4. Correct meaning despite unmatched reference

There were instances where the meaning was conveyed despite being unmatched to the reference. It occurred in two key forms one of which was change in morpheme order:

Reference (as in the Translation platform)	Reference	MT
P33, S2	aba kaa karIM hamanIM	aba hamanI kA karIM
P21, S1	phera trikoNa **paDhIM lA**	phera trikoNa **paDhaba**

Table 7.Correct Meaning for unmatched reference.

An average of such occurrences after considering 10 result pages indicates that such instances have an average occurrence of 38%, i.e., 38 times every 100 sentences.

Another observation in this category includes the presence of words different from the reference but synonymous with it. For example, reference sentence 40/11 : The MT uses the word 'laIkAI' for the reference word 'bachpan' both of which mean childhood or youth. Another example includes the translation of 'ehi' instead of 'IhI' both of which indicate the same meaning. The occurrence of such instances, however, is in not more than 3 out of 11 sentences approximately.

4.5. Translation better than the Reference

An improvement from the previous attempt was indicated in instances of the MT using translated words which were better than those given in the reference. For example, On P41 S10, the word for 'priya' (favorite) in Bhojpuri was translated as **pasandIdA** which is a more local version of the standard reference 'priya'. Other instances include '**bahut sArA'** instead of '**kayiyan' (many), 'bujhAyila'** instead of '**janalas' (came to know).** Such occurrences were between 0-2 times every 20 sentences.

Indication

Four indications can be obtained from the results. First, short sentences are easily and well translated.

Second, the words where inflections were retained were, in many cases, the instances where the word had a closely similar word in Bhojpuri also.

Third, the pattern of verbs was successfully translated at every instance. 'cAha tA' for Sanskrit 'ichchhati' (wants), 'hova tA' for 'bhavati' Sanskrit for 'is happening' were translated with good degree of accuracy.

Fourth, there are no instances of zero matched references. This means that the training succeeded to a great extent. At least one word was definitely translated correctly in the test results obtained after training.

The results of occurrence (every 10 sentences) can be summarized as in the table below:

S.No.	Type	Occurrence
1.	Mixed	2-5
2.	Exact Sanskrit words	≤1
3.	Perfect Translation	0-3
4.	Correct Meaning despite unmatched reference	3.8
5.	Translation better than reference	≤1

Table 8. Overall Evaluation of the Results

5.Conclusion

The current work is first of its kind work in the area MT for Sanskrit to Bhojpuri using conversational corpus by the same authors.

Use of conversational corpora enabled the presence of short sentences . This enabled easy manual translation as well as better alignment.

Similarly, the rise in BLEU score indicated that repetition of reference sentences to the system actually aids in better training.

On such a use, mixed translations which conveyed the overall meaning well emerged as the highest occurring results. Training with better set of data will be able to conclude in higher number of perfect translations.

Further work must be initiated in the direction with more data and larger corpora. Use of simple split-prose text for creation of additional parallel aligned data would enhance the results . Additionally, literary sources with metaphorical use of language may be included in the training data for wider coverage.

Acknowledgements

The authors of this work based their work on the MTHub platform of Microsoft. They are thankful for having been given the platform. Also, Dr. Sankara, Dr. Himanshu Pota and Prof. Baldevanand Sagar, and Bhagini Manjushree helped immensely in the

collection of Sanskrit sentences for which the authors extend their deepest gratitude to them.

References

Behera Pitambar, Neha Maurya and Vandana Pandey. "Dealing with Linguistic Divergences in English-Bhojpuri MT". In: *Proceedings of the South and Southeast Asian Natural Language Processing.* 2016: Osaka, Japan. Pages 103-113. Accessed on: ResearchGate

Maurya, Neha. "Complex Predicate in Bhojpuri". Presented at the 33rd AICL organized by Punjab University, Patiala.

_____. "Classification of Bhojpuri Adverbials" Presented at 9th ICOSAL organized by Punjab University, Patiala.

_____. "Agreement in Bhojpuri." Presented at 11th ICOSAL at BHU, Varanasi.

_____. "Bhojpuri Annotation." Presented at regICON organized by IIT-BHU: Varanasi, 2015.

Pandey, Rajneesh. *Sanskrit-Hindi Statistical Machine Translation: Perspectives and Problems.* Ph.D. Thesis: JNU New Delhi, 2015.

Singh, Srishti. *Challenges in Automatic POS Tagging of Indian Languages: A Comparison of Hindi and Bhojpuri POS Tagger.* M.Phil. Thesis: JNU New Delhi, 2015.

_____ and Esha Banerjee. "Annotating Bhojpuri Corpora using BIS scheme." In: *Proceedings of the WILDRE Conference,* Iceland: 2014.

Sinha, Shagun. *Translation Issues in Conversational Sanskrit-Bhojpuri Language Pair: An MT perspective.* M.Phil. Thesis (Unpublished) : JNU New Delhi, 2016.

Tiwari, Uday Narayan. *The Origin and Development of Bhojpuri Language.* Kolkata: The Asiatic Society, 1960. 2001 Reprint.

Upadhyay, Krishandev. *Bhojpuri Loksahitya.* Varanasi: Vishwavidyalaya Prakashan, 2008.

Vyavaharasahastri. Samskrita Bharati.

Ethnologue website: https://www.ethnologue.com/map/IN_03

Vyavaharsahasri link: http://www.sanskritebooks.org/2009/04/sanskrit-daily-conversation/

YouTube Videos: https://www.youtube.com/watch?v=HTrDZ_5YXow

https://www.youtube.com/channel/UCwqusr8YDwM-3mEYTDeJHzw

https://www.youtube.com/user/ndtv

https://www.youtube.com/watch?v=-UqDljMymMg

Automatic Identification of Closely-related Indian Languages : Resources and Experiments

Ritesh Kumar[1], Bornini Lahiri[2], Deepak Alok[3], Atul Kr. Ojha[4], Mayank Jain[4], Abdul Basit[1], Yogesh Dawer[1]

[1]Dr. Bhim Rao Ambedkar University, [2]Jadavpur University, [3]Rutgers University, [4]Jawaharlal Nehru University
Agra, Kolkata, USA, Delhi
{riteshkrjnu, lahiri.bornini, deepak06alok, shashwatup9k, jnu.mayank, basit.ansari.in, yogeshdawer}@gmail.com

Abstract

In this paper, we discuss an attempt to develop an automatic language identification system for 5 closely-related Indo-Aryan languages of India – Awadhi, Bhojpuri, Braj, Hindi and Magahi. We have compiled a comparable corpora of varying length for these languages from various resources. We discuss the method of creation of these corpora in detail. Using these corpora, a language identification system was developed, which currently gives state-of-the-art accuracy of 96.48 %. We also used these corpora to study the similarity between the 5 languages at the lexical level, which is the first data-based study of the extent of 'closeness' of these languages.

Keywords: Language Identification, Closely-related languages, Awadhi, Braj, Bhojpuri, Magahi, Hindi, Dialect continuum, Indo-Aryan

1. Introduction

Indo-Aryan is the largest and also one of the well-studied language families in the Indian subcontinent. At the same time, it also presents one of the most controversial classification and grouping of languages, especially in terms of languages and their varieties. This is effected by two different reasons. One is the difficulty in tracing the historical path of the Indo-Aryan languages (see Masica, 1993 for detailed discussion on the problem areas and Grierson, 1931; Chatterjee, 1926; Turner, 1966; Katre, 1968; Cardona, 1974 and Mitra and Nigam, 1971 for their somewhat incompatible classification of Indo-Aryan genealogy). The second and more immediate reason is the imposition of Modern Standard Hindi (MSH) over what is now popularly known as 'Hindi Belt' and what has historically been established as a rather complex dialect continuum, with several languages and varieties being spoken in different domains of usage (see Gumperz, 1957, 1958 for an exposition of the different levels of language spoken in the area; King, 1994; Khubchandani, 1991, 1997 for a discussion on the process and resultant of this imposition; also Deo, 2018 for a brief but excellent discussion of issues surrounding the Indo-Aryan languages). Masica (1993) had established the boundaries of this continuum as starting from the language group Rajasthani on the Western side (spoken in the Western state of Rajasthan) to the language group Bihari on the Eastern side, covering other languages like Awadhi, Braj and Bhojpuri. However, with the introduction of MSH as the standard, the situation has become rather complex in the region. As Deo (2018) puts it,

"This top-down state-imposed linguistic norm of Modern Standard Hindi (MSH) has had far-reaching effects on the dialectal situation in the Hindi belt. Until constitutional sanction for Hindi, there were relatively few native speakers of this language in either its deliberately crafted literary version, or its dialectal base, Khari Boli."

However, with MSH being propagated and imposed as the standard through education, media, etc, there has been an increase in both the 'monolingual' speakers of MSH (largely urban, educated population, who no longer speak their parents' language) and 'bilingual' (or even multilingual) speakers who speak one of the languages of the region (with / without an understanding that it is a 'non-standard' variety of MSH) as well as MSH in different domains of usage (Khubchandani, 1997). In addition to this, it must also be noted that the speakers of MSH do not actually speak the same variety of MSH; rather they generally speak some kind of a 'mixed' variety which borrows heavily from their regional language (notwithstanding whether they speak that language or not) and it is these which could actually be called 'varieties' of MSH (see Kumar, Lahiri and Alok, 2013, forthcoming for discussion of one such variety of MSH).

Given this, there are 2 major motivations for working on the languages of the 'Hindi Belt'. The first motivation is technological. Since most of the speakers in the belt are bi-/multi-lingual (in MSH or a variety of it and at least one other language), the actual language usage is marked by code-mixing and code-switching among these languages / varieties. Now even for the most basic task like building a corpus from social media requires that these languages be automatically recognised since there might be a few users who would be writing in Bhojpuri or Magahi and others who might be writing in MSH (or one of its varieties). The second motivation is more theoretical / linguistic. We would like to explore the hypothesis about 'dialect continuum' as well as look at 'similarity' / 'closeness' of the different 'discrete' languages / varieties in this continuum using a data-based approach and give empirical evidence for or against the hypothesis that these languages form part of a 'dialect continuum'.

In this paper, we take into consideration 5 languages of the continuum – Braj, Awadhi, Bhojpuri, Magahi and MSH. Braj is spoken in Western Uttar Pradesh, Awadhi is spoken in Eastern / Central Uttar Pradesh, Bhojpuri in Eastern Uttar Pradesh and Western Bihar and Magahi is spoken in South / Central Bihar. MSH or one of its varieties is now spoken across the belt but it has its dialectal base in Khari Boli which is spoken in Western Uttar Pradesh. Thus in this group, MSH, assuming it to be a variety of Khari Boli, is the Westernmost language, followed by Braj and Magahi is the Easternmost language, with these languages also forming a continuum (shown in Fig 1 below). We will discuss the development of corpus for each of these languages and also give a basic analysis of lexical similarity among these languages. We also discuss the development of a baseline automatic language identification system for these 5 languages using the above-mentioned corpus.

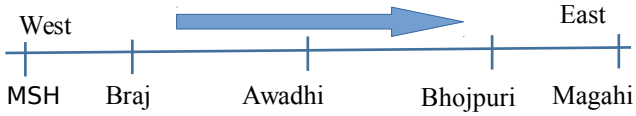

Fig 1 : Position of the 5 languages in the continuum

2. Related Work

Language Identification was generally considered a solved problem with several classifiers for discriminating between languages performing almost perfectly, when trained with word and character-level features. However, in the past few years, inability to replicate similar results in discriminating between varieties and closely-related languages has opened up new questions for the field and kickstarted fresh attempts at solving this problem.

Ranaivo-Malançon (2006) presents one of the first studies that tries to discriminate between two similar languages – Indonesian and Malay – using a semi-supervised model, trained with frequency and rank of character trigrams, lists of exclusive words and the format of numbers (which differed in the two languages in the sense that Malay used decimal points whereas Indonesian uses commas).

Ljubešić et al. (2007) worked on the identification of Croatian texts in comparison to Slovene and Serbian, and reports a high precision and recall of 99%.They made use of a 'black' list of words that increases the performance of the system significantly. This method was further improved by Tiedemann and Ljubešić (2012) improved this method and applied to Bosnian, Croatian and Serbian texts.

Zampieri and Gebre (2012) identified two varieties of Portuguese (Brazilian and European). They used journalistic texts for their experiments and their system gave an impressive 99.5% accuracy with character n-grams. A similar method was later used for classifying Spanish texts that used part-of-speech features, along with character and word n-grams for classification (Zampieri et al., 2013).

Xu, Wang and Li (2016) discusses the development of a system for 6 different varieties of Mandarin Chinese spoken in Greater Chinese Region. They trained a linear SVM using character and word n-grams and also word alignment features. The best system gave an accuracy of 82% which could be explained by the fact that some of the dataset that they used was noisy and also the fact that these are varieties of the same language and expected to be very close to each other, resulting in a difficulty in discriminating among them.

More recently, there have also been an increase in studies focussing on language identification on social media, specially Twitter (Williams and Dagli, 2017; Castro, Souza and de Oliveira, 2016; Radford and Gallé, 2016; Ljubešić and Kranjčić, 2015).

Ljubešić and Kranjčić (2015) worked on 'user-level' language identification instead of 'tweet-level' in which they reached an accuracy of ~98% using a simple bag-of-words model with word unigram and 6-grams and character 3-grams and 6-grams, while classifying 4 very similar South-Slavic languages – Bosnian, Croatian, Montenegrin and Serbian.

Radford and Gallé (2016) makes use of both the language as well as graph properties of tweets for discriminating between 6 languages of tweets – Spanish, Portuguese, Catalan, English, Galician and Basque – and achieved a best score of 76.63%.

Castro et al. (2016) tries to discriminate between Brazilian and European Portuguese on Twitter. They use an ensemble method with character 6-grams and word uni and bigrams to achieve a score of 0.9271.

Williams and Dagli (2017) uses geo-location, Twitter LID labels and editing by crowdsourcing to quickly annotate tweets with their location and then train a classifier using MIRA algorithm to discriminate Indonesian and Malay in tweets. They achieved an accuracy of 90.5% when trained on 1,600 tweets. Their experiments show the utility of using geo-bounding of tweets based on the location of their posting.

Despite this heightened interest in discriminating between similar languages in the European, Asian and also Arabic context, there is hardly any similar attempts to identify Indian languages. Murthy and Kumar (2006) is the only work that we came across for Indian languages. They developed pairwise language identification system for 9 Indian languages from 2 different language families – Hindi, Bengali, Marathi, Punjabi, Oriya (all from Indo-Aryan language family), Telugu, Tamil, Malayalam and Kannada (all from Dravidian language family). Given the fact that these languages are quite distinct from each other and it was a binary classification task (for each pair of languages), the classifier performed almost perfectly which was at par with most of the other state-of-the-art systems available.

Indhuja et al. (2014) also discusses identification of 5 Devanagari-based languages – Hindi, Sanskrit, Marathi, Nepali and Bhojpuri – using character and word n-grams. Even though they claim that these are similar languages, it is not the case despite the fact that they belong to the same language family and use the same script. The system gives a best performance of 88% which is not at par with the performance of modern language ID systems, mainly because of the approach that they took for solving the problem.

Aside from these, there have been hardly any attempt at automatically identifying languages in a multilingual document in Indian languages. Given the fact that most of the documents produced today are multilingual (and they do not include only one of these major languages), with social media enhancing this challenge by several fold, even for the basic task of automated data collection for Indian languages, it is imperative that automatic language identification systems be developed for Indian languages, especially closely-related languages and varieties. In this paper, we present the first attempt towards automatic identification of 5 closely-related Indian languages.

3. Corpus Collection

As we have already discussed above, the 4 languages that we would be working with, share a very complex and, lot of times, hierarchical relationship with MSH. As such, while we have comparatively huge amount of data available for MSH, there is hardly anything available for the other 4 languages – Braj, Bhojpuri, Awadhi and Magahi – in the written form and virtually nothing in the digitised form. The data collection process for all the 4 languages largely followed a similar methodology.

Even though the four languages mentioned above are not used in education or for official purposes, they have a very rich literary tradition[1]. In order to preserve, promote, publish and popularise literary tradition of these languages, local state governments have set up special bodies for some of these languages. For Magahi and Bhojpuri, there are Magahi Akademi (Magahi Academy) and Bhojpuri Akademi (Bhojpuri Academy) in Patna (the capital of the state of Bihar) and for Braj, there is Braj Bhasha Akademi (Braj Bhasha Academy) in Jaipur (the capital of the state of Rajasthan). Uttar Pradesh Hindi Sansthan (Uttar Pradesh Hindi Institute) performs a similar role for Awadhi. Along with these, some individuals, local literary and cultural groups and language enthusiasts also bring out publications in these languages.

Our data collection process mainly consisted of looking for printed stories, novels and essays either in books, magazines or newspapers, scanning those, running an OCR and finally proofreading the OCRed texts by the native speakers of the respective languages.

Since there is no specific OCR available for these languages, we made use of Google's OCR for Hindi that they provide in the Drive API. Since all the languages used Devanagari, we expected the OCR to give a reasonable accuracy and, barring a few times (which was due to the choice of font instead of the language *per se*), it worked rather well. This method helped us in quickly creating the corpus without the need of typing out the whole text.

In addition to this, we also managed to get some blogs in Magahi and Bhojpuri. We crawled these blogs using an in-house crawler built using Google's API. The data for MSH was also crawled from blogs using the same crawler. More details about the sources of data is given in the following subsections

3.1 Awadhi

As mentioned above, the data for Awadhi has been collected from Uttar Pradesh Hindi Sansthan's Library and other publication houses in Lucknow. At present, we managed to get 3 novels written in modern Awadhi -

- Chandawati
- Nadiya Jari Koyla Bhai
- Tulsi Nirkhen Raghuvar Dhama

The current corpus contains data from these 3 novels.

3.2 Bhojpuri

In its current form, the data for Bhojpuri is crawled from 4 different blogs -

- Anjoria
- TatkaKhabar
- Bhojpuri Manthan
- Bhojpuri Sahitya Sarita

We have also got several short story collections, novels and other literary works from Bhojpuri Akademi at Patna and we are in the process of adding those to the present corpus.

3.3 Braj

The data for Braj was collected from 2 main sources – Braj Bhasha Akademi in Jaipur and Braj Shodh Sansthan Library in Mathura. We got two kinds of printed literature from the two sources. We got the following from the Akademi -

- Modern fictional literature published as novel and short stories.
- Critical essays on literature (including a 12-volume set of books on the history of Braj literature, written in Braj)
- Several volumes of a magazine called 'Brajshatdal' consisting of memoirs, short stories, essays and articles.

We got the following from the library in Mathura

- Religious commentaries and essays published as books.

Our corpus currently contains data from each of these sources in almost equal proportion.

3.4 Magahi

Magahi data is mainly collected from 3 sources (see Kumar, Lahiri and Alok 2012, 2014 for more details) -

- Magahi Akademi at Patna : We got several novels, plays and short stories from the Akademi
- Local fieldwork in the areas of Gaya and Jehananbad : We got volumes of 2 Magahi magazines with essays and stories and also a collection of Magahi folktales.
- Crawling the Magahi blogs : The Magahi blogs mainly consist of original and translated literature in Magahi. It must be mentioned that one of the blogs contain a large dictionary of Magahi with example sentences. We have included these example sentences also in our corpus.

Currently the corpus consists of data from all these sources.

3.5 Modern Standard Hindi (MSH)

There are several corpora already available for MSH (Kumar, 2014a, 2014b, 2012 ; Chaudhary and Jha, 2014 and several others). However, in order to keep the domain same as that of other languages, we collected data from blogs that mainly contain stories and novels. Thus the MSH data collected for this study is also from the domain of literature.

[1]It must be mentioned here that it is this rich literature of these and other neighbouring languages that have been co-opted by MSH as 'Hindi' literature and is now forms part of what is known as the tradition of Hindi literature.

The present statistics for each language is summarised in Table 1

Language	Sentences (approx.)
Awadhi	15,000
Braj	30,000
Magahi	170,000
Bhojpuri	62,000
MSH	30,000

Table 1: Corpus statistics for different languages

4. Lexical Overlap and Distance

As we mentioned above, there have been no prior study on the similarity of these languages. So we wanted to explore it using the data that we had. Also since we wanted to build a language identification system, an exploration into the lexical overlap and similarity of these languages would have helped us predict what might be the most useful and productive way of approaching the problem. The overlap matrix (based on lexical overlap in our corpus) is given in Table 2 below.

	MSH	Braj	Awadhi	Bhojpuri	Magahi
MSH	**31,268**	5,721	4,341	6,441	4,803
Braj	5,721	**23,918**	4,466	5,077	4,195
Awadhi	4,341	4,466	**16,977**	4,209	3,622
Bhojpuri	6,441	5,077	4,209	**24,254**	5,538
Magahi	4,803	4,195	3,622	5,538	**21,791**

Table 2: Lexical Overlap Matrix across the 5 languages

This lexical overlap was calculated using a subset of 10,000 sentences of each language, with a total of 50,000 sentences (adding up to a variable number of unique tokens – represented in the overlap matrix above) from the corpora. As you would notice, the results are largely on the expected lines with languages closer together depicting greater overlap. Barring Bhojpuri which shares maximum overlap with MSH (even though they are not the neighbouring languages in the continuum) and least with Awadhi (which is closest to it), all other languages depict the expected behaviour. For example, Awadhi and Braj depict least overlap with Magahi as they are far apart in the continuum. Similarly, Magahi shares maximum overlap with Bhojpuri, which is closest to it. In general, we think the overlap is pretty high (upto 25% of tokens, at times) given the fact that this was a completely naive calculation that was carried out without any normalization of data. This implies that if the languages share the same root but applies different set of morphemes to those roots (which is quite often the case) then that are still considered non-overlapping. It is only when the tokens exactly match that they are considered overlapping. The same calculation with more sophisticated techniques might result in higher overlap numbers.

In addition to this word-level analysis, we also carried out a more nuanced character-level analysis using Levenhestein Edit Distance between words of the two languages. We calculated the edit distance between every pair of words in every pair of language and averaged those out to calculate an average 'distance' between the two languages. Since we do not have a standard way of calculating distance between two languages, we have taken edit distance as the proxy for that. The results are summarised in the form of a distance matrix in Table 3 below. The top row for each language in the table shows an overall edit distance while the bottom row shows the length-controlled edit distance.

As we could see, in general, the edit distance between any pair of languages is quite high and not very far apart from each other. Despite the averages being not very apart from each other, we could see a general trend of average edit distance increases slightly as they become farther in language continuum. Also like in the case of word overlap, all the languages have greatest edit distance from MSH and among those Magahi has the greatest edit distance from MSH while Braj has the least. Similarly, Awadhi and Braj has the smallest edit distance. If we control for the length of the words such that we calculate the edit distance when the length of the words are equal, the overall edit distance is approximately 1 point lower but the trends are still similar.

	MSH	Braj	Awadhi	Bhojpuri	Magahi
MSH	0	6.792	6.823	6.880	6.987
	0	5.853	5.375	5.493	5.549
Braj	**6.792**	0	6.249	6.323	6.433
	5.853	0	5.302	5.432	5.488
Awadhi	6.823	**6.249**	0	6.347	6.455
	5.375	**5.302**	0	5.447	5.496
Bhojpuri	6.880	6.323	6.347	0	6.518
	5.493	5.432	5.447	0	5.481
Magahi	**6.987**	6.433	6.455	6.518	0
	5.549	5.488	5.496	5.481	0

Table 5 : Average edit distance across the 5 languages

5. Language Identification: Experiments and Results

We use a total dataset of 10,000 sentences in each of MSH, Braj, Bhojpuri and Magahi and 9,744 sentences in Awadhi, taken from the corpora discussed above, for developing a sentence-level language identification systems for the 5 languages. We divide the dataset into train:test ratio of 80:20. The train set is used for training a Linear SVM classifier using 5-fold cross-validation. We tune only C hyperparamter of the classifier and arrive at the best classifier using Grid Search technique. We use scikit-learn library (in Python) for all our experiments.

Based on the results obtained in previous studies and their robustness in language identification tasks, we experimented with the most basic frequency distribution of character and word n-gram features for the problem.

Character n-gram features: We used character bigram (CB), trigram (CT), four-grams (CF) and five-grams (CFI) and their different combinations in our experiments

Word n-gram features: We used word unigrams (WU), bigrams (WB) and trigrams (WT) and their combinations for our experiments.

Combined features: We also experimented with different combination of both of the above features.

We used the frequency of each feature as the feature values.

Features	Precision	Recall	F1	Accuracy
Character n-gram features				
CB+CT (C1)	0.96	0.96	0.96	95.868
CB+CT+CF (C2)	0.96	0.96	0.96	96.422
CB+CT+CF+CFI (C3)	0.96	0.96	0.96	**96.482**
Word n-gram features				
WU (W1)	0.79	0.78	0.79	78.361
WU+WB (W2)	0.8	0.79	0.79	**79.167**
WU+WB+WT (W3)	0.8	0.79	0.79	78.744
Combination of character and word n-gram features				
C1+W1	0.96	0.96	0.96	95.878
C1+W2	0.96	0.96	0.96	95.848
C1+W3	0.96	0.96	0.96	95.827
C2+W1	0.96	0.96	0.96	96.372
C2+W2	0.96	0.96	0.96	96.362
C2+W3	0.96	0.96	0.96	96.331
C3+W1	0.96	0.96	0.96	96.472
C3+W2	0.96	0.96	0.96	**96.483**
C3+W3	0.96	0.96	0.96	96.472

Table 4 : Performance of different feature sets on test set

The performance of the different models over the test set is summarised in Table 4.

As we could see, a combination of character bi-gram to 5-grams give the best result of 96.48%. However, the word n-gram features do not seem to work at all. Individually, they give a below par accuracy of 79.16%. This was anticipated and could be explained by the fact that these languages share a large amount of their vocabulary – upto 25% even when going by the strictest evaluation in terms of word-forms – and so they may not act as discriminating features for the classification task. On the other hand, we have also seen that the average edit distance between the words of two languages is close to 7, thereby, implying a greater dissimilarity at character-level, which might be helpful in classification task. Also a combination of character bigrams to 5-grams prove to be most useful, which is consistent with the previous experiments for different languages. As is evident, combining character n-grams with word n-grams does not lead to a better result, probably, because character n-grams already capture the discriminating features of word n-grams.

Table 5 below gives a summary of the language-wise performance of the classifier on the test set. It seems that Magahi is the best-performing language while MSH is the worst one with almost 3 point decrease in F1 score. Awadhi shows the worst precision and best recall which implies that further optimization on Awadhi might lead to a better result.

	Precision	Recall	F1	Test Samples
MSH	0.96	0.95	**0.95**	1996
Braj	0.97	0.95	0.96	1976
Awadhi	**0.94**	**0.98**	0.96	1986
Bhojpuri	0.98	0.97	0.97	1995
Magahi	0.98	0.97	**0.98**	1969

Table 5: Language-wise performance

The confusion matrix in Table 6 shows that misclassification of MSH and Braj as Awadhi is the major source of low precision of Awadhi. Similarly, MSH is misclassified as Braj and Awadhi while Braj is also often misclassified as MSH. What is noticeable is that languages that are far apart in the continuum are *not* confused with each other. So for example, there is hardly any instance of Braj, Awadhi or even MSH being misclassified as Magahi while Bhojpuri is more often taken as Magahi by the classifier (and it holds also the other way round). Moreover, Bhojpuri, Braj and Awadhi are all classified as MSH quite significant number of times. This is despite the fact that Bhojpuri is not at all close to Hindi in the dialect continuum. This could be indicative of the greater influence of MSH over the languages across the so-called 'Hindi Belt' and a slow convergence of these languages into MSH,

which has resulted in greater 'closeness' of all the languages with MSH.

	MSH	Braj	Awadhi	Bhojpuri	Magahi
MSH	1895	**33**	**42**	16	10
Braj	**35**	1887	**46**	2	6
Awadhi	19	20	1943	2	2
Bhojpuri	**23**	8	11	1930	**23**
Magahi	5	5	**16**	**25**	1918

Table 6 : Confusion Matrix on the test set

A closer look at the errors made by the system reveal that quite a few of these errors are because of the noise in the test set. Some are errors because of named entities – strictly speaking this cannot be classified as an error since names are shared across the languages. However, there are some errors which are because of the sufficient overlap between the two languages or availability of sufficient discriminating features in the test sample as in the following example -

1. उ कतल करे जानें, तब्बो फूल बरसावे ले जनता

Even when he keep on killing, the public will still adore him. [Predicted: **Magahi**; Actual: **Bhojpuri**]

2. पत्रिका में अच्छा जीवन चरित सुरेश दुबे 'सरस' जी के लिखे के बात तय होल।

The good life skecth in the magazine was decided after the writing of Suresh Dubey 'Saras' ji [Predicted: **Bhojpuri**; Actual: **Magahi**]

3. अभी बहुत काम है ।

Now there is lot of work left to be done [Predicted: **Awadhi**;Actual: **Hindi**]

4. खैर युद्ध के अवशेष पिंकी देवी बाहर फेंकि आईं ।

Anyway, Pinkey Devi threw out the remains of the war. [Predicted: **Hindi**; Actual: **Awadhi**]

Most of the time, the verbal endings provide a strong clue towards the actual language of the sentence. However, in the examples, given above either the verb is missing (e.g. 1) or it is shared with the other languages (e.g. 2 and 3). In example 3, the sentence is quite short one and so there is not enough discriminating feature available for the classifier. Example 4 can be explained by he use of borrowed words like 'युद्ध' and 'अवशेष', coupled with a verb which is common in Hindi led to it being classified as Hindi.

A lot of these errors could be handled with the use of language-specific morphological features as well as by removing the noise in the data (which, in any case, is not large in number). However, it will remain a difficult task to classify the language in the case of really small sentences or a large amount of overlapping lexicon.

6. Summing Up

In this paper, we have discussed the creation of a comparable corpus of 5 closely-related Indian languages – MSH, Braj, Awadhi, Bhojpuri and Magahi. It need not be mentioned that not only language resources and technologies but even published material (digital / online or in print) in these languages are very scarce and it is a very resource-intensive process to create corpora for these languages. Moreover it is also not possible to crawl data from the web or social media because often these languages are mixed together and it would require manual intervention to segregate those. Based on the corpus developed, we have presented a basic analysis of lexical overlap and average edit distance of these languages, which might be useful from a variationist / sociolinguistic point of view. The analysis shows that there is indeed a greater distance in between languages that are geographically apart, thereby, providing an empirical evidence of a dialect continuum. At the same time, it also provides a strong argument against positing these languages as a variety of one language – they are sufficiently different from each other to be posited as distinct, discrete points in the continuum. We have also developed a baseline automatic language identification system, which is the first such attempt for closely-related Indian languages. The system currently gives an accuracy of over 96% with character 5-grams and may be taken as a baseline for future experiments towards automatically discriminating among these languages.

7. Bibliographical References

Cardona, George. (1974). The Indo-Aryan languages. *Encyclopedia Britannica* (15th edn.). 9: 439-50.

Castro, Dayvid, Ellen Souza, and Adriano LI de Oliveira. (2016). Discriminating between Brazilian and European Portuguese national varieties on Twitter texts. In Proceedings of 5th Brazilian Conference on Intelligent Systems (BRACIS), pages 265 – 270,

Chatterjee, S.K. (1926). The Origin and Development of the Bengali Language, 3 vols. London: George Allen and Unwin.

Chaudhary, Narayan and Girish Nath Jha. (2014). Creating multilingual parallel corpora in Indian languages. In Zygmunt Vetulani (editor), Human Language Technology, pages 527 – 537. Berlin: Springer Verlag.

Deo, Ashwini. (2018). Dialects in the Indo-Aryan landscape. In John Nerbonne, Dominic Watt, and Charles Boberg eds. *Handbook of Dialectology*, Oxford: Wiley-Blackwell (to appear).

Grierson, George A. (1931). On the modern Indo-Aryan vernaculars. *Indian Antiquary*

Gumperz, John T. (1957). Language Problems in the Rural Development of North India. *The Journal of Asian Studies*, 16(2):251–259.

Gumperz, John T. (1958). Dialect Differences and Social Stratification in a North Indian Village. *American Anthropologist*, 60:668–682.

K, Indhuja, Indu M, Sreejith C and P.C. Raghu Raj. (2014). Text Based Language Identification System for Indian Languages Following Devanagiri Script. *International Journal of Engineering Research & Technology*, 3(4):327-331

Katre, S. M. (1968). Problems of reconstruction in Indo-Aryan. Simla: Indian Institute of Advanced Study.

Khubchandani, Lachman. (1991). India as a sociolinguistic area. *Language Sciences*, 13:265–288.

Khubchandani, Lachman. (1997). Indian diglossia. In *Revisualizing boundaries: a plurilingual ethos*. New Delhi: Sage

King, Christopher. (1994). One Language, Two Scripts: The Hindi Movement in Nineteenth Century North India. Delhi: Oxford India.

Kumar, Ritesh. (2012). Challenges in the development of annotated corpora of computer-mediated communication in Indian Languages: A Case of Hindi. In the Proceedings of 8th International Conference on Language Resources and Evaluation (LREC 2012), pages 299 - 302 Istanbul, Turkey.

Kumar, Ritesh. (2014a). Politeness in Online Hindi Texts: Pragmatic and Computational Aspects. Unpublished PhD. Thesis, Jawaharlal Nehru University, New Delhi.

Kumar, Ritesh. (2014b). Developing Politeness Annotated Corpus of Hindi Blogs. In Proceedings of 9th International Conference on Language Resources and Evaluation (LREC 2014), pages 1275 – 1280, Reykjavik, Iceland.

Kumar, Ritesh, Bornini Lahiri and Deepak Alok. (2012). Developing a POS tagger for Magahi: A Comparative Study. In Proceedings of 10th Workshop on Asian Language Resources, pages 105 – 113, Mumbai, India.

Kumar, Ritesh, Bornini Lahiri and Deepak Alok. (2013). Bihari Hindi as a Mixed Language. In Proceedings of Language Contact in India: Historical, Typological and Sociolinguistic Perspectives, pages 199 – 208, Pune India.

Kumar, Ritesh, Bornini Lahiri and Deepak Alok. (2014). Developing LRs for Non-scheduled Indian Languages A case of Magahi. In Zygmunt Vetulani (editor), Human Language Technology, pages 491 – 501. Berlin: Springer Verlag.

Kumar, Ritesh, Bornini Lahiri and Deepak Alok. (2018). Descriptive Study of Eastern Hindi: A mixed language. In Shailendra Singh (editor), Linguistic Ecology of Bihar, New Delhi: Lakshi Publishers (in press).

Ljubešić, Nikola, Nives Mikelic, and Damir Boras. (2007). Language identification: How to distinguish similar languages? In Proceedings of the 29th International Conference on Information Technology Interfaces, pages 541-546, Croatia.

Ljubešić, Nikola and Denis Kranjčić. (2015). Discriminating between closely related languages on Twitter. *Informatica*, 39(1): 1 = 8.

Masica, Colin P. (1993). The Indo-Aryan Languages. Cambridge: Cambridge University Press.

Mitra, A. and R. C. Nigam. (1971). Grammatical Sketches of Indian Languages with Comparative Vocabulary and Texts', Language Monographs (1961 Series), No. 2, Census of India Publication.

Murthy, Kavi Narayana and G Bharadwaja Kumar. (2006). Language identification from small text samples. *Journal of Quantitative Linguistics*, 13(01):57–80.

Radford, Will and Matthias Gallé. (2016). Discriminating between similar languages in Twitter using label propagation. arXiv preprint arXiv:1607.05408.

Ranaivo-Malançon, Bali. (2006). Automatic identification of close languages - case study: Malay and Indonesian. *ECTI Transactions on Computer and Information Technology*, 2:126–134.

Turner, Ralph L. (1966). A comparative dictionary of the Indo-Aryan languages (CDIA L) . vols. 1-2. London: Oxford University Press; vol. 3 (Addenda and Corrigenda) London: SOAS.

Tiedemann, Jörg and Nikola Ljubešić. (2012). Efficient discrimination between closely related languages. In Proceedings of COLING 2012, pages 2619–2634, Mumbai, India.

Williams, Jennifer and Charlie K. Dagli. (2017). Twitter Language Identification Of Similar Languages And Dialects Without Ground Truth. In Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 73–83, Valencia, Spain.

Xu, Fan, Mingwen Wang, and Maoxi Li. (2016). Sentence-level dialects identification in the Greater China region. *International Journal on Natural Language Computing (IJNLC)*, 5(6): 9 – 20.

Zampieri, Marcos and Binyam Gebrekidan Gebre. (2012). Automatic identification of language varieties: The case of Portuguese. In Proceedings of KONVENS2012, pages 233–237, Vienna, Austria.

Zampieri, Marcos, Binyam Gebrekidan Gebre, and Sascha Diwersy. (2013). N-gram language models and POS distribution for the identification of Spanish varieties. In Proceedings of TALN2013, pages 580–587, Sable d'Olonne, France.