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).

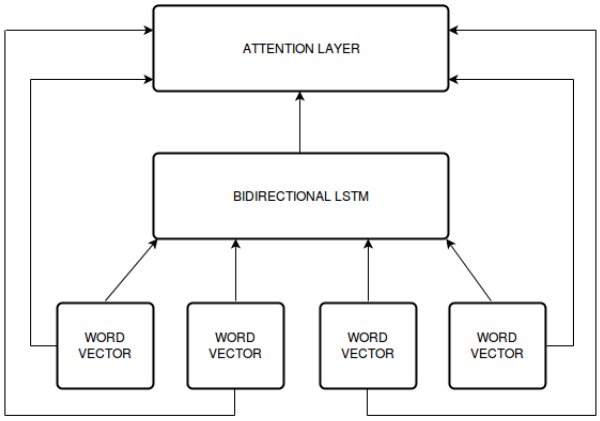

Figure 1: LSTM Model architecture

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.

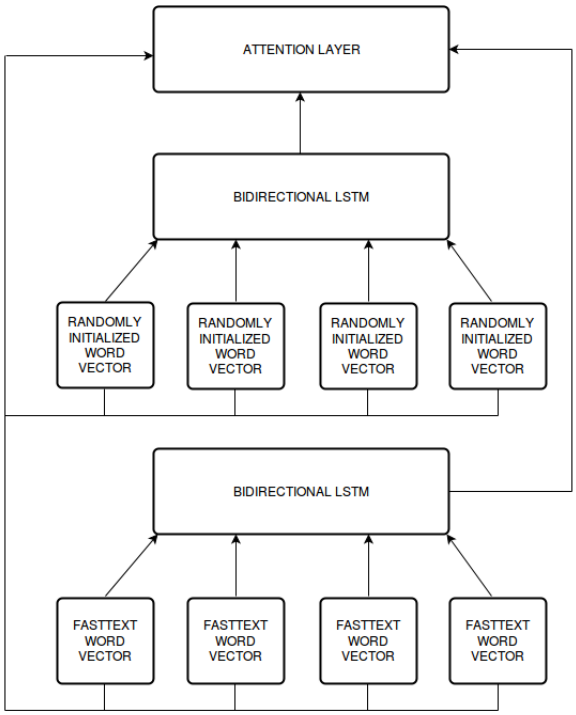

Figure 2: Multi-Channel LSTM architecture

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.