

Cross-Lingual Task-Specific Representation Learning for Text Classification in Resource Poor Languages

by

Narendra Choudhary, Rajat Singh, Manish Shrivastava

in

*27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence
(IJCAI-ECAI-2018 (Humanizing AI Workshop))*

Report No: IIIT/TR/2018/-1



Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2018

Cross-Lingual Task-Specific Representation Learning for Text Classification in Resource Poor Languages^{*}

Nurendra Choudhary¹, Rajat Singh¹, Manish Shrivastava¹

¹ Language Technologies Research Center, Kohli Center on Intelligent Systems,
International Institute of Information Technology, Hyderabad, India.
{nurendra.choudhary,rajat.singh}@research.iiit.ac.in, m.shrivastava@iiit.ac.in

Abstract

Neural network models have shown promising results for text classification. However, these solutions are limited by their dependence on the availability of annotated data.

The prospect of leveraging resource-rich languages to enhance the text classification of resource-poor languages is fascinating. The performance on resource-poor languages can significantly improve if the resource availability constraints can be offset. To this end, we present a twin Bidirectional Long Short Term Memory (Bi-LSTM) network with shared parameters consolidated by a contrastive loss function (based on a similarity metric). The model learns the representation of resource-poor and resource-rich sentences in a common space by using the similarity between their assigned annotation tags. Hence, the model projects sentences with similar tags closer and those with different tags farther from each other. We evaluated our model on the classification tasks of sentiment analysis and emoji prediction for resource-poor languages - Hindi and Telugu and resource-rich languages - English and Spanish. Our model significantly outperforms the state-of-the-art approaches in both the tasks across all metrics.

1 Introduction

Text classification is among the primary challenges in natural language processing. Several approaches, ranging from rule-based systems to machine learning algorithms, have been studied to tackle the problem. Deep learning approaches, especially, show exceptional effectiveness in the task. In this paper, we propose a unified architecture to harness resource-rich languages for the task of representation learning of a resource-poor language for a given task. The proposed architecture consists of two Bi-LSTM networks with mutually shared parameters consolidated by a contrastive loss function. The adopted energy function suits discriminative training for energy based models [LeCun and Huang, 2005].

^{*}This work was presented at 1st Workshop on Humanizing AI (HAI) at IJCAI'18 in Stockholm, Sweden.

The model initiates with a simple one-hot representation based on character trigrams in the sentences. The shared parameters of the siamese network in conjunction with the similarity metric (adopted for the loss function) learn representations of the sentences in accordance to the task. The contrastive loss function projects sentences into the problem space such that sentences with similar tags are closer and different tags are farther from each other.

The input sentences are language-agnostic as the representations only depend on the similarity between their tags. But we require that the tags of both the sentences should represent the same concept. This feature helps us in improving the performance of our model on resource-poor languages by leveraging the abundant resources for the same task available in other languages.

2 Related Work

Distributional semantic vectors [Mikolov *et al.*, 2013] partially capture the semantics of a sentence but ignore syntactic information and diverse word senses. [Mukku *et al.*, 2016] utilizes a combination of semantic vectors and morphology analyzer to solve multi-class sentiment analysis in Telugu language.

Matrix Vector Recursive Neural Networks (MV-RNN) [Socher *et al.*, 2012] provides a solution that considers both individual meaning of a word and its semantic relation with other words in the sentence. The model assigns a vector and a matrix to each word which represent its semantic value and relation with other words respectively. Another line of research [Joshi *et al.*, 2010; Balamurali *et al.*, 2012] benefits from the defined grammatical rules and vocabulary of Hindi language. Although, these solutions are highly accurate, they are susceptible to incorrect spellings, rare words or any exceptional structures frequently observed in informal social media data such as reviews and tweets.

For multilingual emoji prediction, Bi-LSTM model [Bambieri *et al.*, 2017] utilizes recurrent neural networks to capture the character and word sequences of the sentence. The model presents an effective approach to capture the sequence and content of the sentence. Also, it captures close proximity semantic relations in the sentence. However, training the model requires immense data. Thus, the system fails in case of inadequate data.

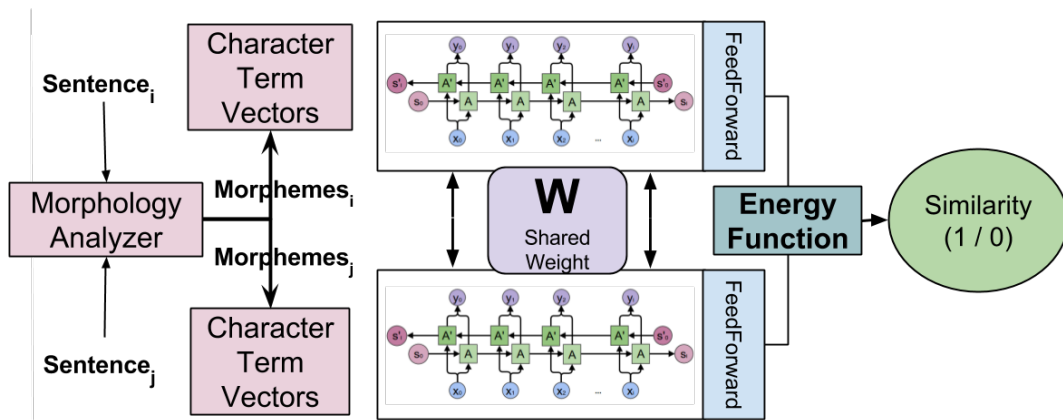


Figure 1: Architecture of the Model

For informal data, methods that require immutable words are ineffective. Given their proven effectiveness [Dhingra *et al.*, 2016; Chung *et al.*, 2014; Vinyals *et al.*, 2015; Bahdanau *et al.*, 2014; Dyer *et al.*, 2015; Wang *et al.*, 2016], we adopt Bi-LSTMs based on sequences of character trigrams. This approach solves the problems of incorrect spellings, rare words and agglutination (in case of Hindi and Telugu).

Although, Bi-LSTMs project sentences into the problem space, we also need the sentences with similar properties closer and those with different properties farther in the problem space. To solve this problem, we utilize siamese networks.

Siamese networks are capable of learning a similarity from given data without requiring specific information about the classes. [Bromley *et al.*, 1994] introduced siamese networks for the task of signature verification. Recently, [Das *et al.*, 2016] utilized siamese networks to solve community question-answering.

3 Architecture of the Model

As depicted in figure 1, the architecture consists of a Bi-LSTM pair connected to a dense feed-forward layer at the top. The Bi-LSTMs capture the sequence and content of the character trigrams in the sentence and project them into the problem space. The contrastive loss function combines the similarity and the sample’s label. Back-propagation through time computes the loss function’s gradient with respect to the weights and biases shared by the sub-networks. The weights are then updated to rectify the error in the epoch.

3.1 Primary Representations

Informal data consists of spelling errors, rare words and variations of the same word. The style of writing a word may also convey a feature (e.g; “Hiiii” conveys a positive sentiment whereas “Hi” is a neutral sentiment). Hence, we consider character trigrams to embed the sentence instead of words. This approach handles spelling errors and rare words. Character trigrams take the information of all the word’s inflections, thus, eliminating the problem of agglutination. Also, this method captures the sentiment of different writing styles

as information is attained on a character-level. To further address the problem of agglutination in morphologically rich languages, we add a morphology analyzer that segments the words into its constituent morphemes. The approach represents a sentence using a one-hot vector with number of dimensions equal to the number of unique character trigrams in the training dataset.

We input character-based term vectors of both the languages’ sentences and a label to the twin networks of our model. The label indicates whether the samples are nearer or farther to each other in the problem space. For positive samples (nearer in the problem space), we feed the twin networks with term vectors of sentences with the same tag. For negative samples (far away from each other in the problem space), we feed the twin networks with term vectors of sentences with different tags.

3.2 Bi-directional LSTM Network

We map each sentence-pair into $[l_i^1, l_i^2]$ such that $l_i^1 \in \mathbb{R}^m$ and $l_i^2 \in \mathbb{R}^n$, where m and n are the total number of character trigrams in both the languages respectively.

Bi-LSTM model encodes the sentence twice, one in the original order (forward) of the sentence and one in the reverse order (backward). Back-propagation through time [Boden, 2002] calculates the weights for both the orders independently. The algorithm works in the same way as general back-propagation, except in this case the back-propagation occurs over all the hidden states of the unfolded timesteps.

We, then, apply element-wise Rectified Linear Unit (ReLU) to the output encoding of the BiLSTM. ReLU is defined as: $f(x) = \max(0, x)$. The choice of ReLU simplifies back-propagation, causes faster learning and avoids saturation.

The architecture’s final dense feed forward layer converts the output of the ReLU layer into a fixed length vector $s \in \mathbb{R}^d$. In our architecture, we empirically set the value of d to 128. The overall model is formalized as:

$$s = \max\{0, W[fw, bw] + b\} \quad (1)$$

where W is a learned parameter matrix (weights), fw is the forward LSTM encoding of the sentence, bw is the backward LSTM encoding of the sentence, and b is a bias term.

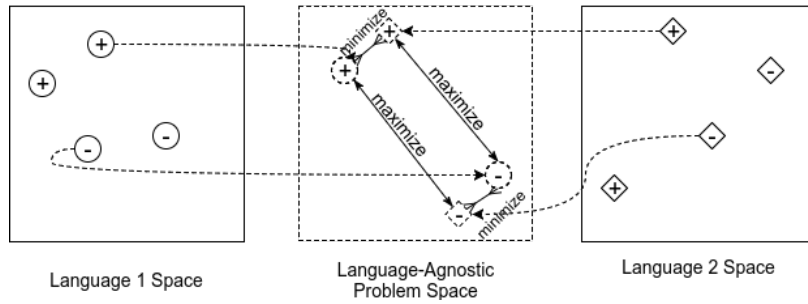


Figure 2: Projection of sentences in both the languages into a common problem space. Distance between similar samples is minimized and dissimilar samples in maximized.

Lan	😊	❤️	😍	😄	😞	😟	😓	😔	😬	😇	😏	😭	😡	😏	😞	😓	❤️	💙	💜
Eng	17.1	15.3	10.0	5.7	4.9	4.7	3.8	3.7	3.6	3.5	3.3	3.2	3.1	3.0	2.8	2.7	2.7	2.7	2.6
Spa	9.7	10.8	13.1	3.1	2.7	6.5	6.3	6.8	6.0	3.4	6.1	4.0	4.6	5.4	3.4	2.8	3.4	3.4	2.0
Hin	9.7	6.8	7.5	3.9	2.8	9.5	4.9	1.5	6.7	2.9	4.3	9.4	8.6	7.7	7.2	1.9	2.3	2.3	2.6
Tel	22.7	5.7	16.6	1.4	0.3	10.8	3.9	0.3	3.0	1.6	0.9	13.4	5.1	6.7	4.2	2.3	0.9	0.5	0.5

Table 1: Distribution of emojis in languages’ tweets. ¹

3.3 Training and Testing Phases

We train our model on the pairs of sentences in both the languages to capture the similarity in their classes. Our architecture differs from other deep learning counterparts due to its property of parameter sharing. Training the network with a shared set of parameters not only reduces the number of parameters (thus, save many computations) but also ensures that the sentences of both the languages project into the same space (shown in figure 2). We learn the network’s shared parameters to minimize the distance between the sentences with the same classes and maximize the distance between the sentences with different classes.

Given an input l_i^1, l_i^2 where l_i^1 and l_i^2 are sentences from both the languages respectively and a label $y_i \in \{-1, 1\}$, the loss function is defined as:

$$loss(l_i^1, l_i^2) = \begin{cases} 1 - \cos(l_i^1, l_i^2), & y = 1; \\ \max(0, \cos(l_i^1, l_i^2) - m), & y = -1; \end{cases} \quad (2)$$

where m is the margin that decides the distance by which dissimilar pairs should be moved away from each other. It generally varies between 0 to 1. The loss function is minimized such that pair of sentences with the label 1 (same class) are projected nearer to each other and pair of sentences with the label -1 (different class) are projected farther from each other in the problem space. The model is trained by minimizing the overall loss function in a batch. The objective is to minimize:

$$L(\Lambda) = \sum_{(l_i^1, l_i^2) \in C \cup C'} loss(l_i^1, l_i^2) \quad (3)$$

where C contains the batch of same sentiment sentence pairs and C' contains the batch of different sentiment sentence pairs. Back-propagation through time (BPTT) updates the parameters shared by the Bi-LSTM sub-networks.

For testing, we randomly sample a certain number (100 in our case) of sentences for each class R_{class} from the language

corpus with higher amount of data. For every input, we then apply the trained model to get the similarity between the input and all corresponding R_{class} . Finally, we select the R_{class} with the most matches with the input as the correct tag.

In case the correlated data available for both the languages are not annotated, we utilize one language’s abundant resources to construct a state-of-the-art sentiment analysis model. The sentiment analysis model in conjunction with the correlation data obtained from our model aids the resource-poor language’s prediction.

4 Experiments

This section studies and evaluates the performance of our model. We also compare our model with the previous approaches in the domain. To perform a proper comparative study and evaluation, we test our system on the specific applications of text classification - multilingual sentiment analysis and multilingual emoji prediction.

4.1 Datasets

We consider varied datasets for both the problems of sentiment analysis and emoji prediction. We also differentiate between languages on the basis of resources. Hindi and Telugu have significantly less resources compared to English and Spanish. Hence, Hindi and Telugu are considered resource-poor languages here and English and Spanish are considered resource-rich. The datasets considered for the task of sentiment analysis are given in Table 4. The datasets for sentiment analysis are annotated into 3 sentiment classes - positive, neutral and negative. The sentiment tags’ distribution in these datasets is given in table 2. The twitter-emoji datasets considered for the task of emoji prediction are given in Table 3.

¹The hearts in the table are of different colors. The most frequent one is *red*, the second most frequent one is *blue* and the last one is *purple heart*

Datasets	Sentence Length	Positive	Negative	Neutral
English - Movie Reviews	429	38%	24%	38%
English - Twitter	12	29%	26%	45%
Spanish - Twitter	14	48%	13%	39%
Hindi - Reviews	15	33%	31%	36%
Telugu - News	13	27%	27%	46%

Table 2: Distribution of the datasets considered for the sentiment analysis task.

Dataset	English	Spanish	Hindi	Telugu
# of tweets	500,000	100,000	15000	6000

Table 3: Datasets for emoji prediction task.

Dataset	# of Sentences
English-Movie Reviews [Pang and Lee, 2005]	5006 movie reviews
English-Twitter [Mozetič <i>et al.</i> , 2016]	103035 tweets
Spanish-Twitter [Mozetič <i>et al.</i> , 2016]	275589 tweets
Hindi-Product Reviews [Mogadala and Varma, 2012]	1004 product reviews
Telugu-News Corpus [Mukku <i>et al.</i> , 2016]	1644 news lines

Table 4: Datasets for sentiment analysis.

Tweets from the ids given by [Barbieri *et al.*, 2017] are used to create English dataset. The dataset consists of the tweets with 18 most frequent emojis in English. Each tweet considered in the datasets consists of a single or multiple instances of a single type of emoji. Table 1 demonstrates the distribution of the emojis in the datasets for emoji prediction task.

4.2 Baselines

The approaches vary based on the language in consideration. Hence, we accordingly define the baselines below. English, Japanese and Spanish enjoy the highest share of data on Twitter². We consider English and Spanish because of their script and typological similarity (both are Subject-Verb-Object).

Average Skip-gram Vectors (ASV): We train a Word2Vec skip-gram model [Mikolov *et al.*, 2013] on a corpus of 65 million raw sentences in English and 20 million raw sentences in Spanish. Word2Vec provides a vector for each word. We average the words’ vectors to get the sentence’s vector. After obtaining each message’s embedding, we train an L2-regularized logistic regression (with ϵ equal to 0.001).

Matrix Vector Recursive Neural Network (MV-RNN): The model [Socher *et al.*, 2012] assigns a vector and matrix to the nodes of a syntactic parsed tree. The vector represents the node’s semantic value and matrix represents its relation with neighboring words. A recursive neural network model is trained using back-propagation through structure to define the nodes’ contribution to the sentence’s sentiment.

Bidirectional LSTM (Bi-LSTM): There are two approaches

²The Many Tongues of Twitter - MIT Technology Review

Method	Accuracy	Precision	Recall	F-score
ASV	52.59%	0.49	0.52	0.50
MV-RNN	79.0%	0.77	0.75	0.76
DSC-T (RF)	67.17%	0.67	0.66	0.66
MNB-H	62.14%	0.61	0.58	0.59
SSC(Eng-Eng)	82.25%	0.83	0.80	0.81
SSC(Spa-Spa)	81.5%	0.83	0.80	0.81
SSC(Hin-Hin)	70.2%	0.72	0.69	0.70
SSC(Tel-Tel)	69.2%	0.70	0.69	0.69
SSC(Hin-Eng)	80.5%	0.82	0.79	0.80
SSC(Tel-Eng)	80.3%	0.82	0.79	0.80

Table 5: Comparison between different language pairs of our model and previous methodologies for sentiment analysis experiment. Siamese Sentiment Classifier (SSC) refers to our model.

- word based and character based Bi-LSTM embeddings. We model the architecture as described in [Barbieri *et al.*, 2017]. The Bi-LSTMs capture the features of the sequence of characters/words into vectors. We, then, use these feature vectors for classification of the sentences.

Hindi and Telugu are the 3rd and 17th most spoken language in the world respectively. But they hold a relatively low share of data. This also further translates to a limited availability of annotated corpus for these languages. The baselines for these languages are:

Domain Specific Classifier (Telugu) (DSC-T): We train a Word2Vec model on a corpus of 700,000 Telugu sentences provided by Indian Languages Corpora Initiative. We train a Random Forest (given by [Mukku *et al.*, 2016]) on the Telugu News dataset to construct our baseline for Telugu language.

Multinomial Naive Bayes Model (Hindi) (MNB-H): We train a multinomial naive bayes model (given by [Sarkar and Chakraborty, 2015]) on the Hindi Review dataset to form our baseline for Hindi language.

The model of MV-RNN is defined for the problem of sentiment analysis, so we do not consider them for the task of emoji prediction. Similarly, Bi-LSTM model is only compared in the emoji prediction task and not sentiment analysis.

4.3 Experimental Setup

We validate our models on the tasks of sentiment analysis and multilingual emoji prediction. Both of these setups are evaluated separately.

Sentiment Analysis

The experiment is a text classification task. We take the English and Hindi sentiment datasets (Eng-Hin) and align each Hindi sentence with English sentences of the same sentiment

Method	5			10			18		
	P	R	F1	P	R	F1	P	R	F1
ASV	0.59	0.60	0.59	0.44	0.47	0.45	0.32	0.34	0.35
Bi-LSTM(W)	0.61	0.61	0.61	0.45	0.45	0.45	0.34	0.36	0.35
Bi-LSTM(C)	0.63	0.63	0.63	0.48	0.47	0.47	0.42	0.39	0.40
DSC-T(RF)	0.34	0.35	0.34	0.31	0.32	0.31	0.24	0.23	0.23
MNB-H	0.45	0.49	0.46	0.42	0.43	0.42	0.38	0.36	0.37
SSC(Eng-Eng)	0.74	0.73	0.73	0.62	0.60	0.61	0.49	0.54	0.51
SSC(Spa-Spa)	0.71	0.72	0.71	0.58	0.59	0.58	0.42	0.42	0.42
SSC(Hin-Hin)	0.54	0.58	0.56	0.45	0.47	0.46	0.39	0.33	0.35
SSC(Tel-Tel)	0.47	0.49	0.48	0.39	0.41	0.40	0.31	0.35	0.33
SSC(Hin-Eng)	0.68	0.70	0.69	0.52	0.56	0.54	0.46	0.43	0.44
SSC(Tel-Eng)	0.63	0.66	0.64	0.49	0.47	0.48	0.41	0.44	0.42

Table 6: Comparison between different language pairs of our model and previous methodologies for emoji prediction experiment. 5,10 and 18 are the number of most frequent emojis considered in that experiment. P,R,F1 are the Precision, Recall and F-scores respectively.

(positive samples) and label them 1. Similarly, we also randomly sample the same number of English tweets with different sentiment (negative samples) for each Hindi Tweet and label them -1. Similarly, we repeat the experiment for all the different pairs of datasets possible.

For the comparative study, we train the baselines defined in section 4.2 on the same datasets. In case of resource-rich languages (English and Spanish), the baselines ASV and MV-RNN are trained on their respective sentiment datasets. We train the baseline MNB-H on Hindi sentiment dataset and baseline DSC-T on Telugu sentiment dataset. The baselines of English compare to the (Eng-Eng) pair’s performance. For each of the other languages *lang*, we compare their baselines with the performance (Eng-*lang*) and (*lang-lang*) pair. Table 5 demonstrates the results of sentiment analysis experiments.

Emoji Prediction

We perform contrastive learning on our model using data made by aligning each English tweet with a set of positive Hindi tweet samples (with the same emoji) with label 1 and a set of negative Hindi tweet samples (with different emoji) of the same size with label -1. We conduct this experiment thrice taking 5, 10 most frequent emojis and all the emojis (18 classes). Similarly, we repeat the experiment for all the different pairs of datasets possible.

For an appropriate comparative study, we also train the baselines defined in section 4.2 on the twitter-emoji datasets. In case of resource-rich languages (English and Spanish), the baseline Bi-LSTM model is trained on the respective twitter-emoji datasets. We train the baseline MNB-H on Hindi sentiment dataset and baseline DSC-T on Telugu twitter-emoji dataset. The baselines of English compare to the (Eng-Eng) pair’s performance. For each of the other languages *lang*, we compare their baselines with the performance (Eng-*lang*) and (*lang-lang*) pair. Table 6 demonstrates the results of the emoji prediction experiments.

5 Analysis of Results

5.1 Qualitative Analysis

As we observe from the results (given in tables 5 and 6) of same language pairs (Eng-Eng, Spa-Spa, Hin-Hin, Tel-Tel),

our model outperforms its counterparts significantly on both the tasks despite the amount of data being same. The improvement in performance is also observed for both the tasks. The reason is that our model is centered around the similarity metric. Bi-LSTMs learn representations of the sentences according to the similarity metric. This helps the model in capturing task-specific features along with necessary semantic features. The shared parameters of the network enable the sentences’ projection to the same problem space based on the similarity metric.

5.2 Quantitative Analysis

From the overall results (given in tables 5 and 6), we observe that the model’s performance in both the tasks is directly proportional to the amount of data available in the language. Also, we observe that in the cases where we pair resource-poor languages with relatively resource-rich languages (Eng-Hin, Eng-Tel), there is a significant improvement in performance compared to their monolingual performance. The Bi-LSTMs and their shared parameters allow the system to project sentences into the same problem space. This allows the problem space to be language-agnostic. Hence, the abundant resources of other languages are able to promote significant performance improvements in resource-poor languages.

6 Conclusion

In this paper, we introduce a solution to leverage resource-rich languages and enhance text classification in resource-poor languages. For this, we proposed an architecture that solves the problem by projecting language-pairs into the same problem space. The model employs twin Bi-LSTM networks with shared parameters to capture a task-specific representation of the sentences. These representations are utilized in conjunction with a similarity metric to group sentences with similar classes together. The qualitative analysis showed that our model outperforms the state-of-the-art methodologies with training on same language pairs. The quantitative analysis presented a significant enhancement in the model’s prediction by training the resource-poor language in conjunction with resource-rich language.

References

- [Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [Balamurali *et al.*, 2012] AR Balamurali, Aditya Joshi, and Pushpak Bhattacharyya. Cross-lingual sentiment analysis for indian languages using linked wordnets. *Proceedings of COLING 2012: Posters*, pages 73–82, 2012.
- [Barbieri *et al.*, 2017] Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. Are emojis predictable? *arXiv preprint arXiv:1702.07285*, 2017.
- [Boden, 2002] Mikael Boden. A guide to recurrent neural networks and backpropagation. *the Dallas project*, 2002.
- [Bromley *et al.*, 1994] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a “siamese” time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.
- [Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [Das *et al.*, 2016] Arpita Das, Harish Yenala, Manoj Chinakotla, and Manish Shrivastava. Together we stand: Siamese networks for similar question retrieval. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 378–387, 2016.
- [Dhingra *et al.*, 2016] Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William W Cohen. Tweet2vec: Character-based distributed representations for social media. *arXiv preprint arXiv:1605.03481*, 2016.
- [Dyer *et al.*, 2015] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*, 2015.
- [Joshi *et al.*, 2010] Aditya Joshi, AR Balamurali, and Pushpak Bhattacharyya. A fall-back strategy for sentiment analysis in hindi: a case study. *Proceedings of the 8th ICON*, 2010.
- [LeCun and Huang, 2005] Yann LeCun and Fu Jie Huang. Loss functions for discriminative training of energy-based models. In *AISTATS*, volume 6, page 34, 2005.
- [Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [Mogadala and Varma, 2012] Aditya Mogadala and Vasudeva Varma. Retrieval approach to extract opinions about people from resource scarce language news articles. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, page 4. ACM, 2012.
- [Mozetič *et al.*, 2016] Igor Mozetič, Miha Grčar, and Jasmina Smailović. Multilingual twitter sentiment classification: The role of human annotators. *PloS one*, 11(5):e0155036, 2016.
- [Mukku *et al.*, 2016] Sandeep Sricharan Mukku, Nurendra Choudhary, and Radhika Mamidi. Enhanced sentiment classification of telugu text using ml techniques. In *SAIIP@IJCAI*, pages 29–34, 2016.
- [Pang and Lee, 2005] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics, 2005.
- [Sarkar and Chakraborty, 2015] Kamal Sarkar and Saikat Chakraborty. A sentiment analysis system for indian language tweets. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 694–702. Springer, 2015.
- [Socher *et al.*, 2012] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 1201–1211. Association for Computational Linguistics, 2012.
- [Vinyals *et al.*, 2015] Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781, 2015.
- [Wang *et al.*, 2016] Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. Learning distributed word representations for bidirectional lstm recurrent neural network. In *HLT-NAACL*, pages 527–533, 2016.