Shallow Parser for Telugu in a Low Resource Setting

Murali Manohar[1], Ganesh Katrapati[2], and Dipti Misra Sharma[2]

[1] BML Munjal University, Gurgaon, India
kmanoharmurali@gmail.com
[2] NLP & MT Lab, LTRC, KCIS, IIIT-Hyderabad
{ganesh.katrapati@research.iiit.ac.in},{dipti@iiit.ac.in}

Abstract. In this paper, we address the challenge of building Shallow Parser for a morphologically richer and low resource language, Telugu using variants of Neural Network models. In a low-resource setup, although it is advisable to use Hand-Picked features, rule based methods rather than looking at the little data for training data hungry Neural Networks, we propose that we can make utmost use of limited data present when linguistic supervision is absent. We show that the resource scarcity and high degree of agglutination in Telugu language can be addressed by considering character level representations, Conditional Random Fields in Neural Networks to achieve State-of-the-Art results by 2% than that of the traditional models that leverage morphological features. We also propose a Transfer learning based approach, aiming to alleviate the model to transfer knowledge about the language, learnt in one task, to another.

Keywords: POS Tagging · Chunking · char-word · LSTM.

1 Introduction

POS tagging is the task of assigning Parts of Speech tag for each word in a particular sentence. Neighboring words and their order carry crucial information required for POS tagging. POS tagging is a sequence labeling task. For the same reason, we have used an architecture called sequence2sequence LSTM(Long Short Term Memory), which captures sequential information. One problem being faced by any statistical machine learning or neural networks model is that, during testing phase model may come across a new word that it has never seen during the training phase. This can either be a new word or misspelled word. Any word level model would treat the new word as unknown word, but in sequence labeling tasks, every word is required. So, in order to extract the information from a new word, we use a char-word level model that uses character level information of each word. The motivation behind this approach comes from the fact that Telugu is a morphologically rich language and words carry more information than the context itself. We try to leverage this information to build an efficient POS tagger. Advent of neural networks alleviated the situation of hand picking the features, but the problem lies in its need for abundant annotated data.

Unsupervised methods rely on a resource rich language and Parallel corpora or its equivalent. Pre-trained language model answers the question of having an inbuilt knowledge of the language. Language modeling, has been shown to capture many aspects of language required for downstream tasks, such as long-term dependencies, hierarchical relations, and sentiment analysis. In this paper, we explored the feasibility of leveraging a pre-trained language model to build an efficient POS tagger. We achieved encouraging accuracies without any need for relying on resource rich language or large annotated data. We analyzed the reason behind model not getting better accuracies.

2 Related Work

As Dravidian languages carry high amount of morphological information, learning POS tagging on these languages is a difficult task because of the unavailability of large annotated corpus and high degree of agglutination. [11] used a TnT model by estimating transition and emission probabilities of Kannada using the cross-language Telugu. Whereas, [10] used CRF, TnT models with morphological features to built a Telugu POS Tagger with 91.23 % accuracy.

Neural Network Approaches: [4] used Recurrent Neural Networks(RNNs) for building a POS Tagger, as they help in capturing the sequential information. The main problem of RNNs in the context of POS Tagging is their ability of providing information being limited to previous words but not successive words in the context. This was overcome by Bidirectional RNNs. [12] associated character level representations of each word with its corresponding word-representation to handle the rich morphology and intra-word information which is crucial for tasks like POS tagging. Despite having the intra-word and contextual information, what matters the most is the grammatical nature of a language. In sequence labeling task using RNNs, the probability of the current tag given the previous tag is not considered, which is often important. [3] dealt this by adding a Conditional Random Field layer(CRF)[5] on the top of Bidirectional Long Short Term Memory Networks (LSTM) outputs' layers. CRF layer provides sentence level tagging information.

Transfer Learning: Word embeddings find their position at a shallow level when it comes to language understanding. Although they capture the semantic information, they are used only at the beginning layer of models. Data hungry nature of the Neural Networks, comes from the fact that they need to understand the implicit features of the language and as well as the target task itself. Transfer Learning is widely used in Computer Vision, but it isn't leveraged well in NLP until recently. The traditional transfer learning approach used in Deep learning in NLP is the use of word embeddings. [2] fine-tuned a pre-trained language model [6], to train a sentiment classifier with relatively lesser examples and achieved better accuracies. [9] introduced the concept of ELMo (Embeddings from Language Model), which are obtained from running a Bidirectional Language Model. These embeddings are proven to be perform relatively better than the traditional word embeddings like Glove[8], Word2Vec[7].

3 Corpus details

We have used Multilingual Indian Language Corpora Initiative (ILCI) [1] Hindi-Telugu parallel Agriculture-Entertainment corpus for building the POS tagging dataset [3] and Hindi-Telugu general corpus [4] for building the chunking dataset. BIS tagset[13] is used for tagging the sequences in ILCI corpus. Further details of the corpus are mentioned in the Table 1.

- Models for POS Tagging are trained on 6417 sentences and tested on 1605 sentences, which are randomly shuffled.
- Models for Chunking are trained on 750 sentences and tested on 250 sentences, which are randomly shuffled.

Table 1. ILCI corpus details

Task	Sentences	Tokens	Vocabulary
POS Tagging	8022	103595	23138
Chunking	1000	10370	3342

Word embeddings and the corpus on which they are trained are made public[5]. Details of the word embeddings are shown in the Table 2.

Table 2. Word embeddings details

Description	Value
Scraping Tool	HTML-Boilerplate[6]
Tokenization Tool	Indic Tokenizer[7]
# Sentences	22386073
# Tokens	279577900
Method followed	Word2Vec[8]
Algorithm	CBOW[9]
Context window size	8

[3] Includes parsing and collecting required instances
[4] `http://tdil-dc.in/index.php?option=com\_download\&task=showresourceDetails\&toolid=1270\&lang=en`
[5] `https://drive.google.com/drive/folders/1fEt7aIzYWGQKto3Nt51M5CdjtzxMqdCz`
[6] `https://html5boilerplate.com`
[7] `https://github.com/ltrc/indic-tokenizer`
[8] `https://github.com/tmikolov/word2vec`
[9] `https://github.com/tmikolov/word2vec`

4 Methodology

In this paper, we have explored various neural network models and we are presenting the ones with better accuracies. The whole idea of considering only neural networks is to skip the time taking step of any NLP task, i.e hand picking the features. Along with the supervised approaches, where we plainly input the training data and train the model, we have also tried another approach called Transfer Learning using Pre-Trained Language Model.

Requirement of large annotated data for neural networks comes from the fact that a model needs to understand the irregularities and patterns of a language and then learn the target task. Our transfer learning approach deals with the idea of having the model learned the patterns of the language before hand, thus requiring lesser annotated data to train the model on target task.

The configuration details of LSTMs used in all our experiments except Transfer learning approach are shown in Table 3.

Table 3. Models' Configurations

Description	**Values**
word embedding dimension	200
Word-level LSTM Hidden state dim	100
char-word LSTMs Hidden states dim	50
character embedding dim	50
Loss function	Negative log likelihood
Optimizer	SGD

4.1 Vanilla LSTM POS Tagger

As shallow parsing is a sequence labeling task, we have used a seq-seq model with LSTM to capture sequential information. We have tried various modeling techniques.Bidirectional LSTM serves the purpose of carrying sequential and contextual information. Hence, it is able to perform better than LSTM.

Input: Input to the below mentioned models is a whole sentence, where each word is given to the LSTM neural network at every time step.

4.2 Char-word model

It may happen that the some words may come out of the vocabulary, on which word embeddings aren't trained on. This can be due to misspelled words or new words. Just like an n-gram used for misspelled words, we use a character level LSTM for each word whose output is concatenated with word embedding. This way, even if the word embedding has no information of the new word, the character level LSTM output will carry relevant information required for POS

tagging.

Input: Input is similar to that of the above mentioned models except that a new LSTM is added at character level whose input is a sequence of characters at each time step.

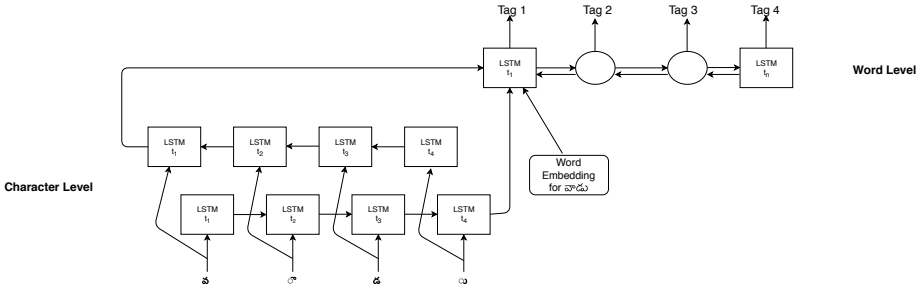

Fig. 1. Representation of char-word BiLSTM

4.3 CRF layer on top of char-word Model

Apart from the word and contextual information obtained by char-word implementation using Bi-Directional LSTM, there is a need to look at sentence level tag information. CRF (Conditional Random Fields) ensures the final predictions are valid by learning required constraints. For the same reason, we have added CRF on the top of char-word model.

Emission and transition scores are the parameters of this model. Emission scores are the output probabilities of each label after BiLSTM layer. The emission score for a word i comes from the hidden state at the time step i.

For Transition scores, a separate matrix is maintained to store the transition scores between all labels. For boosting up the model, we have added "<Start>" and "<Stop>" labels. This matrix is updated with training.

For an input sequence x and output label sequence y, we compute

$$P(y/x) = \frac{exp(Score(x, y))}{\sum_{y'} exp(Score(x, y'))} \tag{1}$$

For every path y given x, *Score(x,y)* is calculated, which is the sum of transmission and emission scores.

$$Score(x, y) = \sum_i \log(Emit(y_i, x_i)) + \log(trans(y_i, y_{i-1})) \tag{2}$$

In the similar way, golden score is also calculated. *Golden score* refers to the *Score(x,y)* calculated for expected label sequence y given an input sequence x.

Loss Function calculated is as follows,

$$Loss = Present\,score - Golden\,score \tag{3}$$

Loss is used to backpropagate and update the transition matrix.

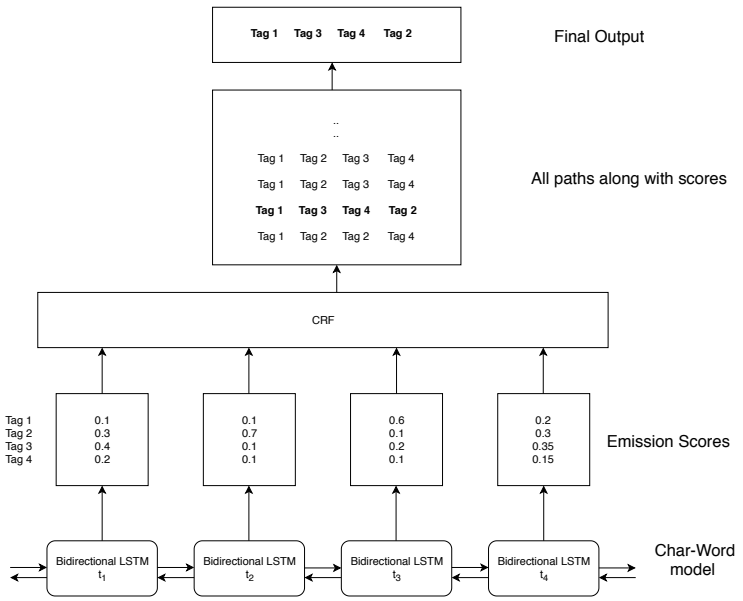

Fig. 2. char-word BiLSTM CRF

4.4 POS input for Chunking

Along with the above two methods for sequence labeling, we have inputted the Parts of Speech Tags information into the model for better chunking by concatenating the char-word level information with their corresponding POS tag information. POS tagging information is given in the form of embeddings, which are randomly initialized and fine-tuned upon training the model. Current approach performs better than all the existing approaches for chunking.

4.5 Transfer Learning

Compared to related unsupervised tasks such as CBOW and Skip gram, language modeling performs better on syntactic tasks even with less training data. We tried if the language modeling approach can be beneficial in POS Tagging task. We have used AWD LSTM language model [6] which is currently the State of the Art with a perplexity of 52 in English language. We pre-trained language

model on a large Telugu corpus [10]. It involves embedding layer followed by three LSTMs with dropouts for every LSTM and a dense layer for predicting the next word. Transfer learning approach is highly inspired from fastai[11]. Concept of Transfer learning comes in, when we detach the last dense layer of pre-trained language model, used for predicting the next word and store the embedding weights & attach a required architecture.

In our case, we remove the last dense layer of the language model that has the next word as the output and apply a time distributed dense layer (*i.e* dense layers with same parameters) to hidden state at every time step of the last(3^{rd}) LSTM of the language model.

Results are not satisfactory when we have tried POS tagging in Telugu with this approach. One reason is that the perplexity of language model isn't coming down. It is settling around 350.

5 Experimental Results

The experimental results of POS tagging, Chunking are shown in the Table 4 & Table 5. The first column talks about the model architecture. 2^{nd} entry reports the test accuracy of the model and 3^{rd}, 4^{th} and 5^{th} entries represent weighted Precision, Recall and F1-score. Baseline CRF has been trained using CRF++ tool[12] with the previous and successive words as features.

Table 4. POS tagging Experimental results

Model	Accuracy	#P	#R	#F
CRF(Baseline)	87.0	0.89	0.87	0.88
LSTM	88.5	0.88	0.89	0.88
BiLSTM	89.8	0.90	0.90	0.90
char-word LSTM	93.2	0.93	0.94	0.93
char-word BiLSTM	93.5	0.93	0.94	0.93
char-word BiLSTM CRF	93.9	0.94	0.94	0.94

The results of POS tagging using transfer learning approach are mentioned in the Table 6. In Seq2Seq model, sequence of inputs will have their corresponding outputs. In Context2Tag, information of the neighboring words (+1 and -1) are provided along with the word to predict its tag.

6 Error Analysis & Observation

It has been observed that agglutinative languages like Telugu carry more information in the words themselves than in the context. Our Baseline proves

[10] https://drive.google.com/drive/folders/1fEt7aIzYWGQKto3Nt51M5CdjtzxMqdCz
[11] http://course.fast.ai/lessons/lesson10.html
[12] https://taku910.github.io/crfpp/

Table 5. Chunking Experimental results

Model	Accuracy	#P	#R	#F
CRF(Baseline)	77.0	0.82	0.78	0.80
char-word LSTM	79.3	0.75	0.79	0.76
char-word BiLSTM	79.9	0.75	0.79	0.76
char-word BiLSTM CRF	87.1	0.87	0.87	0.87
char-word BiLSTM CRF with POS input	98.2	0.98	0.98	0.98

Table 6. POS Tagging through Transfer learning approach's Experimental results

Model	Accuracy	#P	#R	#F
Seq2Seq	79	0.70	0.79	0.74
Context2Tag	84.7	0.83	0.85	0.84

this fact. However, when widened to new domain of text for POS tagging, this approach fails because of unexpected morph forms of the words.

When error analysis is performed on both word and char-word LSTM models, following observations are made:

- Whenever a word is in its morphed form and has no word embedding, then the character level information is helping the model get correct tags.
- Word level model simply assigns the words with no embeddings to highly dominating classes like NN,V_VM_VF,V_VM_VNF. Few examples are shown in Table 7.

Table 7. Word level error examples

Word	Actual tag	Predicted
maralA	RB	V_VM_VNF
mUlarUpaMlo	N_NN	V_VM_VF
411.01	QT_QTC	N_NN
...	...	...

- There are annotation errors in ILCI corpus. Examples shown in Table 8

Table 8. Annotation error examples

Word	Wrongly Annotated Tag
25-30	V_VM_VNF
paluchani	V_VM_VNF
chAlA	N_NN
...	...

- Classes like Auxiliary verbs, Echo words (RD_ECH),V_VM_INF,V_VM,RP_NEG, Reciprocal pronouns, RD_RDF are less occurring in the dataset.
- char-word model fails mostly at predicting NNP. If stem or affixes form a meaningful morpheme, it predicts it as N_NN. While Word LSTM, as mentioned earlier, predicts default Verb or N_NN classes.
- There are rare instances where char-word model is false and word level prediction being true. The instances where char-word model's predictions failed is because it might be giving more priority to the sub words which belong to another class. This is troublesome as well as solving some problems, which is explained below.
  - "alAgaite" was predicted as RB in character level instead of CC_CCD. This mistake was made by character level since "alAge" (Sub word) had RB tag in other instances.
  - For the word "devuni", char-word model is paying more attention to the word "devuDu". "kAraNaMgA" is also facing similar problem with "kAraNaM".
- Spelling error in the beginning of a word has resulted in the failure of proper prediction, which is later resolved by char-word Bi-Directional LSTM.
- Word level LSTM was only giving the past information. When we imparted successive words' information along with past information, there is only 1 percent increase in accuracy which is reassuring the observation made on words carrying more information than the context.
- char-word Bi-LSTM handled spelling errors well as it processes the given sequence in both the directions.
- As Morphological richness results in a vast vocabulary, we have limited the vocabulary size to 200000, which is restricting the capability of building an efficient language model.

Table 9. Generalized error types

Type	**Explanation**
Annotation errors	Manual annotation errors
Design errors	Annotation ambiguities. Ex: Between Gerund Verbs, Nouns and NN & JJ etc. "niMpaDaM" is a Gerund but can also be N_NN.
Attention based errors	Inefficiency in assigning priority to sub-words

Future Work

We would like to extend this work by leveraging a sub-word level representation like syllables & vowels, which is not as sparse as word level and computationally

less expensive than character level representations. As the perplexity of Telugu language model is not coming down, cross-lingual word embeddings along with word alignment tools can be leveraged to build POS taggers with merely no dataset. We'd like to explore sub-word level language model, which can alleviate the situation of parsing remaining Dravidian languages as well. The only part, where char-word model failed is in its inability to assign priority to different parts of a word. Applying Attention mechanism on the character level representation would help in ameliorating the shallow parser.

References

1. Choudhary, N., Jha, G.N.: Creating multilingual parallel corpora in indian languages. In: Vetulani, Z., Mariani, J. (eds.) Human Language Technology Challenges for Computer Science and Linguistics. pp. 527–537. Springer International Publishing, Cham (2014)
2. Howard, J., Ruder, S.: Fine-tuned language models for text classification. CoRR **abs/1801.06146** (2018)
3. Huangrate, Zhihengu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging (08 2015)
4. Juan Antonio, P.O., L.Forcada, M.: Part-of-speech tagging with recurrent neural networks. In: IJCNN'01. International Joint Conference on Neural Networks. Proceedings. IEEE Operations Center, Washington, DC (July 2001)
5. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282–289. ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)
6. Merity, S., Keskar, N.S., Socher, R.: Regularizing and Optimizing LSTM Language Models. arXiv preprint arXiv:1708.02182 (2017)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc. (2013)
8. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. vol. 14, pp. 1532–1543 (01 2014). https://doi.org/10.3115/v1/D14-1162
9. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. CoRR **abs/1802.05365** (2018)
10. Phani Gadde, M.V.Y.: Improving statistical pos tagging using linguistic feature for hindi and telugu. In: Proceedings of ICON-2008: 6th International Conference on Natural Language Processing. Pune, India (December 2008)
11. Reddy, S., Sharoff, S.: Cross language pos taggers (and other tools) for indian languages: An experiment with kannada using telugu resources. In: Proceedings of the Fifth International Workshop On Cross Lingual Information Access. pp. 11–19. Asian Federation of Natural Language Processing, Chiang Mai, Thailand (November 2011)
12. Cicero Nogueira dos Santos, B.Z.: Learning character-level representations for part-of-speech tagging. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. Journal of Machine Learning Research, Beijing, China (June 2014)

13. Vaz, E., Walawalikar, S.V., Pawar, D., Sardesai, D.: Bis annotation standards with reference to konkani language. In: Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (SANLP). pp. 145–152. Mumbai, India (December 2012)