International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 10 (2018) pp. 7857-7867

Fast and Accurate Parts of Speech Tagging for Kannada-Telugu Pair

Chandramma[1], Dr. Piyush Kumar Pareek[2]

[1]Research Scholar, Department of Computer Science and Engineering
East west institute of technology, VTU, Bangalore, India.

[2]Associate Professor, Department of Computer Science and Engineering
East west institute of technology, VTU, Bangalore, India.

Abstract

India is a country of rich divergence of languages. Each language having large speaker base, yet some language have minimal and very limited linguistic information. Like Kannada, Malayalam and Marathi are relatively very poor when compared with its counterparts such as Telugu, Tamil and Hindi. Many Indian language pair's shares similar morphology and syntactic behavior like Malayalam with Tamil, Marathi with Hindi, and Kannada with Telugu. Training a part-of-speech (POS) tagging model using Morphological rich tag set (Telugu) for low-resources languages (Kannada) usually requires linguistic knowledge and resource about the relation among source and the target language. This paper introduce a cross-lingual transfer learning model for Kannada using Telugu resources without ancillary resources such as parallel corpora. In this paper, we present a morphological analyzer for both Kannada and Telugu and construct large corpora. Our model can be extended to build morphological analyzer for other Indian language. Experiment are that to evaluate the performance of proposed POS tagger model for both monolingual and cross-lingual tagger. The study shows the efficiency of proposed approach and is much faster than state-of-art POS tagger model.

Keywords: Condition random field, Hidden Markov model, POS tagging.

INTRODUCTION

Parts-of-speech-tagging is defined as the process of assigning to each word in a sentence, a label which indicates the status of that word within some system of categorizing the words of that language according to their morphological and/or syntactic properties. In many Natural Language Processing applications such as word sense disambiguation, information retrieval, information processing, parsing, question answering, and machine translation, POS tagging is considered as the one of the basic necessary tool. A part-of-speech is a grammatical category, commonly including verbs, nouns, adjectives, adverbs, determiner, and so on. Different pos taggers have been developed for various languages and are used in many different NLP applications.

POS taggers are broadly classified into two categories called rule based and stochastic (or Statistical) [1]. In case of rule based approach applies a set of hand-written rules and uses contextual information to distinguish the tag ambiguity and assign POS tags to words. The main drawback of rule based system is that it fails when the text is unknown. The rule based system cannot predict the appropriate tag. Hence for achieving higher accuracy in this system we need to have an exhaustive set of hand coded rules. The relative failure of rule-based approaches, the increasing availability of machine readable text and the increase in capability of hardware (CPU, memory, disk space) with decrease in cost are some of the reasons, researchers to prefer corpus based POS tagging. The performance of the POS tagging model hugely depends on the corpus data with which it is trained. Stochastic (statistical) taggers are either HMM based, choosing the tag sequence which maximizes the product of word likelihood and tag sequence probability, or cue-based, using decision trees or maximum entropy models to combine probabilistic features. A statistical approach includes frequency and probability. The simplest statistical approach finds out the most frequently used tag for a specific word from the annotated training data and uses this information to tag that word in the unannotated text. The problem with this approach is that it can come up with sequences of tags for sentences that are not acceptable according to the grammar rules of a language.

In most of the languages including the Dravidian language like Kannada and Telugu, ambiguity is one of the key issue that must be addressed and solved while designing a POS tagger. For different context words behave differently and hence the challenge is to correctly identify the POS tag of a token appearing in a particular context. In [2] presented a tag set and architecture for POS tagging in Marathi language based on corpus based and machine learning approaching. However, their model is expensive [3] due to lack of corpora, lexicons or morphological analyzers. The lack of POS tagger is due to very limited work is carried out for a particular language. As a result some language has very limited resource of corpus to build efficient tagger. Recently, cross lingual learning has been presented [4], [5] for building POS tagger from resource poor language (Kannada) using resources from resource rich language (Telugu). This work consider building a cross lingual learning among Kannada and Telugu which is typographically similar.

In [4], [5] built POS taggers for a target language using parallel corpus. The source (cross) language is expected to have a POS tagger. First, the source language tools annotate the source side

of the parallel corpora. Later these annotations are projected to the target language side using the alignments in the parallel corpora, creating virtual annotated corpora for the target language. A POS tagger for the target is then built from the virtual annotated corpora. These approaches are based on Condition Random Field (CRF), and Hidden Markov models (HMM) [6], [7], [8], [9] and [10]. They aim to gain from information shared across languages. The main disadvantage of all such methods is that they rely on parallel corpora which itself is a costly resource for resource-poor languages. As a result our work consider developing a POS tagger for a target language using the resources of another typologically related language for Dravidian language (Kannada and Telugu language pair).

In this paper, we aim to build a cross language learning for Dravidian languages which are as efficient as compared to existing mono-lingual tagger. For performance evaluation, we experiment with the resource-poor language Kannada, by building various cross language part of speech taggers, using typologically-related and resource rich language Telugu. Our part of speech taggers can also be used as a morphological analyzer since our POS tags is composed of morphological information.

The Contribution of research work is as follows:

- Here we added morphological information to the tag set
- Our tagger can also be used as morphological analyzer.
- Achieves higher accuracy and reduction in processing time for POS tagging.
- Performance evaluation proposed tagger over state-of-art monolingual and cross-lingual tagger presented.

The rest of the paper is organized as follows. In section II problem statement are defined. In section III the proposed POS tagging model is presented. In penultimate section experimental study is carried out. The conclusion and future work is described in last section.

PROBLEM STATEMENT

The main issues of state-of-art POS tagging mechanism is that it directly relies on parallel corpora which is a costly affair for resource constraint language. Another issue is to solve ambiguity problem the state-of-art POS tagging mechanism consider context instead of word. Though it aid in improving the accuracy of tagging but it incurs computing overhead for tagging [3]. As a result, is not applicable for future application needs which requires fast and accurate tagging. Cross lingual tagging mechanism is adopted by state-of-art technique for building POS tagger for resource poor language (Kannada) using resources from resource rich language (Telugu). However, they all built considering parallel corpora set which is expensive. As a result, this work considers building POS tagger extracting typological information among languages (Kannada-Telugu pairs. To overcome the research problems an efficient POS tagging model needs to be designed that brings a good tradeoff between accuracy and performance requirement. The proposed Part-of-Speech Tagging for Dravidian language (Kannada-Telugu pair).

PART-OF-SPEECH TAGGING MODELLING FOR DRAVIDIAN LANGUAGE (KANNADA-TELUGU PAIR)

This paper present an accurate, fast and efficient part-of-speech tagging for Dravidian language. Firstly we present data preprocessing steps for building POS tagging. Secondly, this work present an $Trigrams'n'Tags$ (TnT) based part-of-speech tagger for Dravidian language (Kannada-Telugu) applying second order Markov model. The architecture of proposed Cross lingual POS Tagging model is shown in Fig. 1 below.

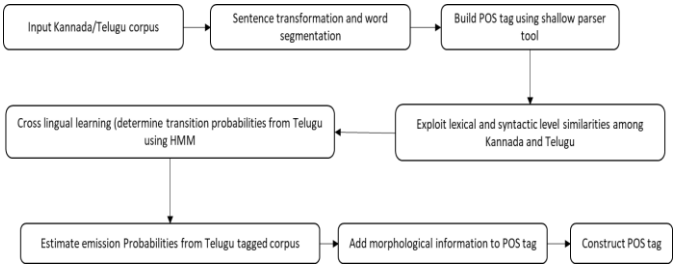

Figure 1. Architecture of proposed Cross lingual POS Tagging

a) POS Tag set:

Fortunately the communities of research scholars working on Indian languages have helped in designing the tag set. We will use the wealth of experience generated by them to finalize the tag set. Our tag set is an adoption of the work proposed by [6], [27] as part of Indian Language Machine Translation (ILMT) project. The tag set listed in Table 1 includes 25 tags covering the different parts of speech of the language. It is designed to take advantage of the machine learning process and also facilitate further NLP processing tasks.

Table 1. POS TagSet

Sl. No.	Tag	Description	Example
1	NN	Noun	ವಸ್ತು
2	NNC	Compound Noun	ಕೆಂಪು ಗುಲಾಬಿ
3	NNP	Proper Noun	ಬೆಂಗಳೂರು
4	NNPC	Compound Proper Noun	ನರೇಂದ್ರ ಮೋದಿ
5	PRP	Pronoun	ಅವನು
6	DEM	Demonstrative	ಆ
7	VM	Verb Finite	ಕುಣಿದನು
8	VAUX	Auxiliary Verb	ಕುಣಿಯುತ್ತಾ
9	JJ	Adjective	ಒಳ್ಳೆಯವ
10	RB	Adverb	ವೇಗವಾಗಿ
11	PSP	Postposition	ಜೊತೆ

Sl. No.	Tag	Description	Example
12	CC	Conjunction	ಮತ್ತು
13	WQ	Question Words	ಯಾಕೆ
14	QF	Quantifiers	ಬಹಳ
15	QC	Cardinal	ಒಂದು
16	QO	Ordinal	ಒಂದನೇ
17	INTF	Intensifier	ತುಂಬ
18	INJ	Interjection	ಛಲೇ
19	NEG	Negative verbs	ಬರೋದಿಲ್ಲ
20	SYM	Symbol	. , ?
21	RDP	Reduplication	ಹೇಳಿಹೇಳಿ
22	UT	Quotative	ಎಂದು
23	NUM	Numbers	೮೪
24	ECH	Echo words	ಹಾಗೆ ಹೀಗೆ
25	UNK	Unknown	Hi

b) Data preprocessing:

The Dravidian language such as Kannada and Telugu makes part-of-speech tagging a challenging task because of its unambiguous characteristics. As a result it is necessary for the input sentences needs to be preprocessed and fed into the part-of-speech tagging model. Here we shows the steps considered for the preprocessing phase for Dravidian corpora:

- All Dravidian corpora of Kannada and Telugu are transformed to have only one sentence per line ending one of the punctuation such as '.', '?', or '!'.
- All zero-width non-breaking have been eliminated.
- Each words are partitioned or chunked with a single space symbol.
- All the adherent words have been tokenized.
- If a word is an adherent with a punctuation sign, symbol, or other characters, it is segmented.

The above steps are considered to build the Dravidian (Kannada-Telugu) corpus to be utilized for part-of-speech tagging. Among them the first two step are common which is performed for all language pairs and part-of-speech models. The next two steps are performed specifically for Dravidian language which is used for distinguishing the word. This steps are crucial to bring a good tradeoff among length of sentence and number of distinctive words. By adopting technique described in [11] the problem of the length of the sentence is decreased. This work considers increasing the length of the sentence and decreasing the number of distinctive words. This setting aid in achieving enhanced performance. The last step used into disjointing non-related characters. If our model do not segment the word and its adjunct punctuation sign, the model will consider them as a whole word, and this is not tolerable/adequate for our model. Since part-of-speech model requires them as a two distinctive words and not as one word. More detail discussion of segmentation process is described in later section of paper. In next sub section c we present a $Trigrams'n'Tags$ (TnT) based part-of-speech tagger for Dravidian language (Kannada-Telugu).

c) Trigrams'n'Tags/TnT based tagger for Dravidian language (Kannada-Telugu):

In this section, we present a $Trigrams'n'Tags$ (TnT) based part-of-speech tagger for Kannada-Telugu pair applying second order Markov model. Applying TnT [13] for POS tagger is as efficient as any other method is shown in [12]. The tagger model is composed of transition and emission probabilities. The states/transition of the model represent tags and outputs/emission represent the words. The transition probabilities depends on pairs of tags, i.e. states. Emission probabilities mainly depends on the most recent category. The tag sequence of given word sequences is chosen by computing

$$\underset{k\ldots k_{\mathcal{K}}}{\operatorname{argmax}}\left[\prod_{m=1}^{\mathcal{K}} \mathcal{P}(k_m|k_{m-1},k_{m-2})P(j_m|k_i)\right]P(k_{\mathcal{K}+1}|k \quad (1)$$

where $k_m \ldots k_{\mathcal{K}}$ are their corresponding POS tags, the extra/supplementary tags k_{-1} , k_0, and $k_{\mathcal{K}+1}$ are beginning and end of sequence markers, and $j_m \ldots j_{\mathcal{K}}$ is the word sequence of length $\mathcal{K}$. Utilization of these supplementary tags aid in improving the tagging results (i.e.) the existing model just stop with a "loose end" at lost word. However, in our approach if sentence boundaries are not marked, our tagger adds these tags if it identifies one of [., ?, !, ;] as a token.

Transition and emission probabilities are computed from a tagged dataset. Firstly, from relative frequencies derivation we use the maximum likelihood (ML) probabilities $\hat{\mathcal{P}}$:

$$\boldsymbol{Unigrams}: \quad \hat{\mathcal{P}}(k_3) = \mathcal{R}(k_3)/_{\mathcal{T}} \quad (2)$$

$$\boldsymbol{Bigrams}: \quad \hat{\mathcal{P}}(k_3|k_2) = \mathcal{R}(k_{2,}k_3)/_{\mathcal{R}(k_2)} \quad (3)$$

$$\boldsymbol{Trigrams}: \quad \hat{\mathcal{P}}(k_3|k_1,k_2) = \mathcal{R}(k_{1,}k_{2,}k_3)/_{\mathcal{R}(k_{1,}k_2)} \quad (4)$$

$$\boldsymbol{Lexical}: \quad \hat{\mathcal{P}}(j_3|j_2) = \mathcal{R}(j_{3,}k_3)/_{\mathcal{R}(k_3)} \quad (5)$$

for all j_3 in the lexicon and $k_{1,}k_{2,}k_3$ in the tag set. $\mathcal{T}$ is the cumulated number of tokens in the training dataset. This work considers or states that if corresponding nominators and denominators are zero then the ML probability to be zero.

Secondly, contextual frequencies are normalized and lexical frequencies are processed by performing words that are not in the lexicon as described in section d.

d) Normalization:

Generation of trigram probabilities from dataset generally cannot be applied directly due to sparse data problem. As a result there are not adequate occurrence for each trigram to dependably compute the probability. Besides, setting a probability to zero has an undesired effects because corresponding trigram may have never occurred in the dataset. As a result, the probability of an entire sequences is set to zero if its usage is needed for a new text sequences, due to which it makes impractical to rank different sequence containing a zero probability.

The normalization parameter that offers best outcome in $Trigrams'n'Tags$ is linear interpolation of $unigrams, bigrams,$ and $trigrams$. Therefore, we compute a $trigram$ probability as follows

$$\hat{\mathcal{P}}(\hbar_3|\hbar_1,\hbar_2) = \beta_1\hat{\mathcal{P}}(\hbar_3) + \beta_2\hat{\mathcal{P}}(\hbar_3|t_2) + \beta_3\hat{\mathcal{P}}(\hbar_3|\hbar_1,\hbar_2) \quad (6)$$

here $\mathcal{P}$ represent probability distributions. Since, $\hat{\mathcal{P}}$ are ML approximations of the probabilities, and $\beta_1 + \beta_2 + \beta_3 = 1$.

In our work the value of βs do not depend on the particular trigram, since we use context-independent technique of linear interpolation. This aids in achieving better outcomes than state-of-art context-dependent technique. One cannot compute a different set of βs for each trigram due to sparse data problem. Hence, we group trigram by frequencies and compute grouped sets of βs. No prior work has considered frequency grouping for linear interpolation in POS tagging based on our knowledge.

The values of β_1,β_2, and β_3 are computed by deleted interpolation. This technique aid in consecutively eliminates each trigram from the training dataset and computed ideal/optimal values for the βs from all other $ngrams$ in the datasets. By identifying the frequency counts for unigram, bigram, and trigram, the weights can be resourcefully established with a computing time linear in the number of different trigrams. The algorithm to compute weight for context independent linear interpolation is shown in Algorithm 1. An important thing to take note of algorithm 1 is subtracting 1 means taking concealed/unknown data into account. This is considered in our work to eliminate over fit the training data for achieving better result.

Algorithm 1: Algorithm for computing the weights for $\beta_1, \beta_2, \beta_3$ considering known $n-gram$ frequencies. $\mathcal{T}$ is the size of the dataset. The outcome of the expression is set to zero, if the denominator in one of that expressions is 0.

Step 1: Set $\beta_1 = \beta_2 = \beta_3 = 0$

Step 2: $\forall$ trigram $\hbar_1,\hbar_2,\hbar_3$ with $\mathcal{R}(\hbar_1,\hbar_2,\hbar_3) > 0$

Subject to the maximal of the following conditions:

Condition $\frac{f(\hbar_1,\hbar_2,\hbar_3)-1}{f(\hbar_1,\hbar_2)-1}$: **Increase** β_3 by $\mathcal{R}(\hbar_1,\hbar_2,\hbar_3)$

Condition $\frac{f(\hbar_2,\hbar_3)-1}{f(\hbar_2)-1}$: **Increase** β_2 by $\mathcal{R}(\hbar_1,\hbar_2,\hbar_3)$

Condition $\frac{f(\hbar_3)-1}{\mathcal{T}-1}$: **Increase** β_1 by $\mathcal{R}(\hbar_1,\hbar_2,\hbar_3)$

Step 3: End

Step 4: End

Step 5: Normalized $\beta_1, \beta_2, \beta_3$

e) Handling of Unknown Words:

Kannada is a highly inflected language with three gender forms, masculine, feminine, neutral or common, and two number forms, singular and plural. The number forms interestingly shows inflection based on the gender, number and tense, of the commodity of reference, among other factors. To handle unknown words for highly inflected language such as Kannada and Telugu suffix analysis is presented [20]. Tag probabilities are set based on the word's ending. The suffix is a strong predictor for word classes. The probability distribution for a specific suffix is produced from all words in training dataset that has similar suffix (i.e. final sequences of character of words) of predefined maximum length. Then the probabilities are normalized by consecutive abstraction.

This computes the probability of a tag $\hbar$ assumed that the last $\hbar$ letters w_m of an t letter word: $\mathcal{P}(\hbar|w_{t-\hbar+1} \dots w_t)$. The sequence of increasingly more common contexts ignores exceedingly characters of the suffix, such that $\mathcal{P}(\hbar|w_{t-\hbar+3} \dots w_t), \dots, \mathcal{P}(\hbar))$ are utilized for normalization which is obtained as follows

$$\mathcal{P}(\hbar|w_{t-m+1} \dots w_t) = \frac{\hat{\mathcal{P}}(\hbar|w_{t-m+1} \dots w_t) + \theta_m\mathcal{P}(\hbar|w_{t-m+1} \dots w_t)}{1+\theta_m} \quad (7)$$

for $m = \hbar \dots 0$, using the ML $\hat{\mathcal{P}}$ from frequencies in the lexicon, weights θ_m and the initialization

$$\mathcal{P}(t) = \hat{\mathcal{P}}(\hbar). \quad (8)$$

The ML approximation for a suffix of length m is expressed from dataset frequencies by

$$\hat{\mathcal{P}}(\hbar|w_{t-m+1}\dots w_t) = \mathcal{R}(\hbar, w_{t-m+1}\dots w_t) \Big/ \mathcal{R}(w_{t-m+1}\dots w_t) \quad (9)$$

In our approach we inverse conditional probabilities $\mathcal{P}(w_{t-m+1}\dots w_t|\hbar)$ due to adoption of Markov model which are obtained by applying Bayesian inversion. A theoretical proved model used the standard deviation of the ML probabilities for the weights θ_m [20]. For achieving that, the following steps should be followed

- Firstly, one has to identify a good parameter for $\hbar$, the longest suffix used. In our tagger approach, $\hbar$ depends on the word in question. This work consider the longest suffix that we can identify in the training set (i.e., for which the frequency is equal to or greater than1), but at most 10 characters which is determined through observation.

- Secondly, similar to contextual weights β_m, we adopt context-independent approach to determineθ_m. It considered to be an optimal strategy to set all θ_m to the standard deviation of the unconditioned ML probabilities of the tags in the training dataset which is set as follows

$$\theta_m = \frac{1}{\mathcal{u}-1}\sum_{n=1}^{\mathcal{u}}\left(\hat{\mathcal{P}}(\hbar_j) - \bar{\mathcal{P}}\right)^2 \quad (10)$$

for all $m = 0 \dots \hbar - 1$, using a tagset of $\mathcal{u}$ tags

and the mean

$$\bar{\mathcal{P}} = \frac{1}{\mathcal{u}}\sum_{n=1}^{\mathcal{u}}\hat{\mathcal{P}}(\hbar_n) \quad (11)$$

This parameter will usually be range of $0.03 \dots 0.10$.

- Thirdly, this paper consider different approximation for lower and upper cases words, i.e., two different suffix tries are kept based on the capitalization of the word. Since our model consider POS tagging using cross lingual corpus for varied languages. This aid in enhancing the tagging performance outcome.

- Lastly, to handle unknown word this work consider suffix handling to words with a frequency lesser than predefined threshold. Based on observation we have considered the threshold size to 10. Since unknown word are predominantly infrequent. As a result rather than using suffix of frequent words we consider using suffixes of infrequent words in the lexicon. This aid in better approximation for unknown words.

f) Capitalizations:

Our approach consider POS tagging using cross lingual corpus for varied languages. As a result, we need to address the disambiguation issue. To address the disambiguation for different language and tag sets capitalization information is considered to be a useful information. In state-of-art model tags are generally not informative of capitalization, but probability distribution around non-capital word are different from those of capitalized. This effect is significantly higher for English corpus and very smaller for Dravidian languages. For that, this work use Boolean function $\mathcal{v}_m$ that are false if $\mathcal{j}_m$ is not a capitalized word and true otherwise. These Boolean function are added to the contextual probability distributions. Rather than

$$\mathcal{P}(\hbar_3|\hbar_1, \hbar_2) \quad (12)$$

We consider

$$\mathcal{P}(\hbar_3, \mathcal{v}_3|\hbar_1, \mathcal{v}_1, \hbar_2, \mathcal{v}_2) \quad (13)$$

and equations (3) to (5) are updated accordingly. This is proportional to doubling the size of the tag set and utilizing diverse tags depending on capitalization.

g) Sequence Search/Decoding phase:

Finding the sequence of POS tags with the highest probabilities along the path for a given sequence of words is called decoding [21]. The state-of-art model adopts Viterbi algorithm [22] for guaranteed performance in finding states with highest probability. However, it incur high processing time. To minimize this work introduce beam search. In [23] [24] empirically showed adoption of beam search aid in performance improvement without affecting accuracy. This work considers that each state that obtains a φ parameter smaller than the largest φ divided by some threshold parameter is omitted from further processing. Nonetheless, for real-world case the right choice of θ is obtained. Empirically, a value of $\theta = 1000$ turned out to approximately double the speed of the tagger without affecting the accuracy [23] [24]. The processing speed mainly depends on the ratio of unknown words and on the mean ambiguity rate. The generic beam search pseudocode [23] [24] is given below

Pseudocode for Beam search algorithm

```
Function DECODE(sent, agenda):
  CLEAR(agenda)
  ADDITEM(agenda, "")
  for index in [0..LEN(sent)]:
    for cand in agenda:
      new←APPEND(cand, sent[index])
      ADDITEM(agenda, new)
      for pos in TAGSET():
        new ←SEP(cand, sent[index], pos)
        ADDITEM(agenda, new)
    agenda←N-BEST(agenda)
  return BEST(agenda)
```

h) Implementation Process of POS tagging for Dravidian language (Kannada-Telugu):

This work aims to constructing an efficient, fast and accurate Dravidian language (Kannada and Telugu) part-of-speech tagger using Eq. (1). We use *Trigrams'n'Tags* for POS tagging as described in section c and computes transition and emission probabilities of Kannada using the Telugu. Since our tag set has both morphological and POS data encoded in it. Adoption of Markov model aid in using morphological data for POS tagging and similarly the POS tag can be used to obtain morphological information. The steps considered for our work are as follows

- **Step 1:** firstly, creation of large dataset of Kannada and Telugu is done.
- **Step 2:** Secondly, establishing the transition probabilities of Kannada and Telugu using *Trigrams'n'Tags* described in section c on Kannada and Telugu dataset downloaded using step 1 respectively. We also consider that the transition probabilities of Kannada and Telugu to be same. Since Kannada and Telugu are syntactically identical.
- **Step 3:** Thirdly, compute the emission probabilities of Kannada from machine annotated Kannada or Telugu dataset.
- **Step 4:** lastly, use the transition and emission probabilities obtained using step 2 and 3 to build a part-of-speech tagger for Kannada.

i) Creation of Kannada-Telugu datasets:

Dataset collection is considered to be slow and expensive. However, with the growth of Web the dataset collection can be automated [14], which aid in reducing time and cost of dataset collection [15]. The following steps is considered for automated collection of Kannada and Telugu dataset creation.

- Firstly, the automated data collection techniques requires a frequency list specific for language of interest for initialization of data collection process. The frequency list of language is constructed using [25] and all HTML markup links are removed from raw corpus dataset that need to be extracted. Then the frequency list is constructed from tokenized Wikipedia dataset. Then, we consider only top 500 and 5000 words of frequency list as the high $\mathbb{H}$ and mid frequency $\mathbb{M}$ ones which are used as seed word for the dataset collection.
- Secondly, we generate arbitrary queries. Here we consider 10,000 arbitrary queries of word size of two and no duplication query are considered (i.e. no identical queries are considered). The algorithm to determine the best query length for each language is shown below in Algorithm 2.

Algorithm 2: Best Query Length

Step 1. set t = 1 (number of words)

Step 2. create 100 queries using t seeds per query

Step 3. Sort queries based on the number of hits they obtain.

Step 4. Identify hit count for 90th query ($min - hits - count$)

Step 5. if ($min - hits - count$) $<$ 10 get $t - 1$

Step 6. $t = t + 1$, go to step 2

- Thirdly, each query is sent to Google or Microsoft Bing search engine and pages respective to the hits are downloaded from web and are converted it into UTF-8 encoding. Then, these pages are cleaned to remove irrelevant blogs and links and extract only plain text. We used simple language model to remove spam and irrelevant language (i.e., apart from Kannada and Telugu). We download pages where it satisfies $\mathbb{H}$ and $\mathbb{M}$ frequency hits or else it is omitted. Still there will be presence of duplication [6] of some pages. To eliminate such data model defined in Algorithm 3 is used. Finally we will be able to obtain clean corpus dataset for Kannada and Telugu. In similar way we can obtain corpus data for other Dravidian and other language. The algorithm identify and eliminate duplicate page is shown below

Algorithm 3: Identify and remove duplication

Step 1. Sort the content names by their content sizes and store all the content names in a list.

Step 2. Classify first 500 non duplicate content (traversing linearly on content names list) using [26].

Step 3. Compare rest of the contents, a content at a time, with these 500 non-duplicate contents

Step 4. Remove any duplicate contents found and store the rest of the content names in next_contentnames list

Step 5. filenames = next_contentnames

Step 6. Continue from **step 2**.

j) Transition Probabilities computation for Dravidian language (Kannada-Telugu):

Here we compute the transition probabilities of Kannada. Transition probabilities are probabilities of transition to a state from the previous states. Here each state denotes a tag and therefore represented as $\mathcal{P}(\mathcal{k}_m|\mathcal{k}_{m-1}, \mathcal{k}_{m-2})$. This work then compute transition probabilities considering two ways as follows

- Firstly, this work compute transition probabilities using Telugu (source) and the transition probabilities among tags are likely to be same. Since both languages are typologically/syntactically same. This work consider the transition probabilities of Telugu to be nearly identical to the transition probabilities of Kannada. This work used publicly available annotated corpus from [17] to tag the Telugu dataset downloaded using section *i*. The tagged dataset is then converted and then using *Trigrams'n'Tags* described in section *c* we compute transition probabilities.

- Secondly, this work annotated the Kannada dataset obtained using section *i* using existing tagger. Then we compute the transition probabilities from the machine annotated Kannada dataset. This process is considered to evaluate the performance of cross-lingual based approach over monolingual based approaches. Our approach is efficient, fast and accurate. Since we are able to predict unknown words using morphological information aiding. As a result aid in achieving better part-of-speech prediction.

k) Emission probabilities computation for Dravidian language (Kannada-Telugu):

Here we compute the emission probabilities of Kannada for a given language. Transition probabilities are probabilities of transition to a state from the previous states. Here emission correspond to words and state to tag and therefore represented as $\mathcal{P}(\mathcal{j}_m|\mathcal{k}_m)$. This work then compute emission probabilities considering two ways as follows

- For computing emission probabilities for cross lingual corpus requires bilingual dictionary or parallel corpora. Since Telugu and Kannada are mutually understandable [18], this work exploit the lexical similarities among Kannada and Telugu.
- Firstly, a Telugu lexicon is constructed by training *Trigrams'n'Tags* described in section *c* on the Telugu dataset obtained using section *i*. The lexicon has the information related to Telugu word and its respective part-of-speech tags along with their frequencies. Then we construct word list for Kannada corpus and are encoded in ASCII format. Post completion of word list construction string matching is performed with Telugu corpus which is also ASCII encoded. For building lexicon for Kannada we consider that Kannada words, its tags and their frequencies are identical to the most similar Telugu words. Then lexicon is constructed for Kannada with each word having its probable tags and frequencies established from Telugu and this lexicon is utilized to compute transition probabilities.
- Secondly, for each morphological set from annotated Telugu dataset, we establish all its probable fine-grain part-of-speech tags. Then we assign all the tags applicable for each word in Kannada, as learned from Telugu uniformly based on its morphology determined by morphological analyzer. However, it increase search space. As a result incur computation overhead.

- Thirdly, though we learn tags adopting existing part-of-speech tagger. As a result we do not use emission probabilities of existing tagger, since we do not use information about tag frequencies. The state-of-art tagger is just utilized to construct lexicon for Kannada. The automated corpora creation presented here aided in creating large corpora. As a result, when running tagger on large data corpus, our lexicon contains most of Kannada words and their respective part-of-speech tag and morphological information. This lexicon address the issues of [17] and aid in achieving an efficient and fast tagger. Our model can tags the unknown word even it is absent in lexicon based on *Trigrams'n'Tags* transition probabilities. Thus aid in reducing the search space and improves speed of tagging.

- Lastly, we compute emission probability of annotated Kannada corpus using exiting tagger to evaluate the performance comparison with cross-lingual tagger when transition probabilities are obtained from Telugu. We also evaluate performance of the monolingual tagger when transition probabilities are obtained from Kannada. Our Monolingual tagger for Kannada is robust, fast and accurate when compared with state-of-art techniques.

In next section performance evaluation of proposed POS tagger is evaluated considering different dataset/corpus and performance parameter over state-of-art technique. Experiment of proposed POS tagger is evaluated over CRF and Markov model are experimentally shown.

RESULT AND ANALYSIS

This section present performance evaluation of proposed POS tagging over state-of-art model. Performance is evaluated in term of accuracy of tagging and computation time. The POS tagger methods was experimented on sentences taken from various articles from Wikipedia. These sentences consisted of 80000 words, out of which first 64000 were taken as training data and the remaining 16000 words were taken as test data. The test data contained 40 % of words that were not part of training data. The proposed POS tagger tool is implemented using Dot net framework 4.5 using C#, Perl and Python programming language and the experiment are conducted on 64-bit I-5 class Intel processor, 16 GB RAM with dedicated Nvidia CUDA enabled GPU. The POS tagger tool tagged 16000 test data and the results are presented below in Fig. 2 and Fig. 3 in term of

accuracy achieved. The comparison result is tabulated in Table II and Table III shows, proposed POS tagger achieves significant performance over stat-of-art techniques. Our model took about 60000 milliseconds to tag 16,000 word whereas the exiting model took about 190000 milliseconds which is shown in Fig. 4. and Table IV. A 68.43% computation overhead reduction is achieved by proposed approach over existing model [7]. The significant speedup of tagging of proposed model is due to adoption of beam search for performing decoding.

Table II. Experimental comparison over state-of-art monolingual technique

Experiment Id	**[28]**	**[27]**	**[7]**	**[30]**	**[29]**	**Proposed**
Language used	Kannada	Kannada	Kannada	Kannada	Sinhala	Kannada
Cross lingual support	No	No	No	No	No	Yes
Algorithm	CRF	CRF and HMM	n-gram CRF	SVM	Hybrid	HMM
Decoding algorithm used	-	-	-	Viterbi	Viterbi	Beam
Number of words	1000	54,000	16,000 (80,000)	54,000	25,087	16000 (80,000)
Number of Tags used	36	25	36	30	-	25
Accuracy	55%	84.58%	92.94%	86.0%	70.38%	94.62%

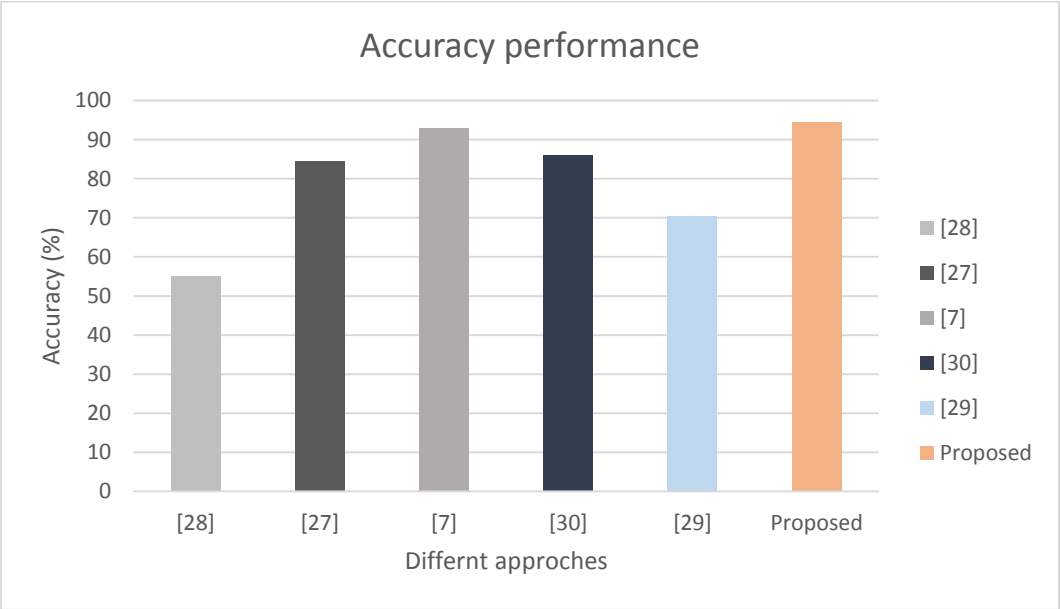

Figure 2. Accuracy performance of proposed POS Tagging and over state-of-art monolingual technique

Table III. Experimental comparison over state-of-art cross lingual technique

Experiment Id	**[31]**	**[4]**	**[32]**	**[33]**	**Proposed**
Source Language pair	English	English	Multilingual	English	Kannada-Telugu
Cross lingual support	Yes	Yes	Yes	Yes	Yes
Multi-lingual transferring learning support	Yes (14)	Yes (11)	Yes (18)	Yes (9)	Yes (1)
Parallel corpora required for training	No	Yes	No	Yes	No
Algorithm	BLSTM	Discriminative learning using NN	Character-level recurrent neural network	UMPOS	HMM
Decoding algorithm used	-	-	-	-	Beam
Number of words	1280	>80	-	1M	16000
Number of Tags used	32-96	35-79	-	-	25
Accuracy	91.24 %	95.75%	93.7	78.52	94.62%

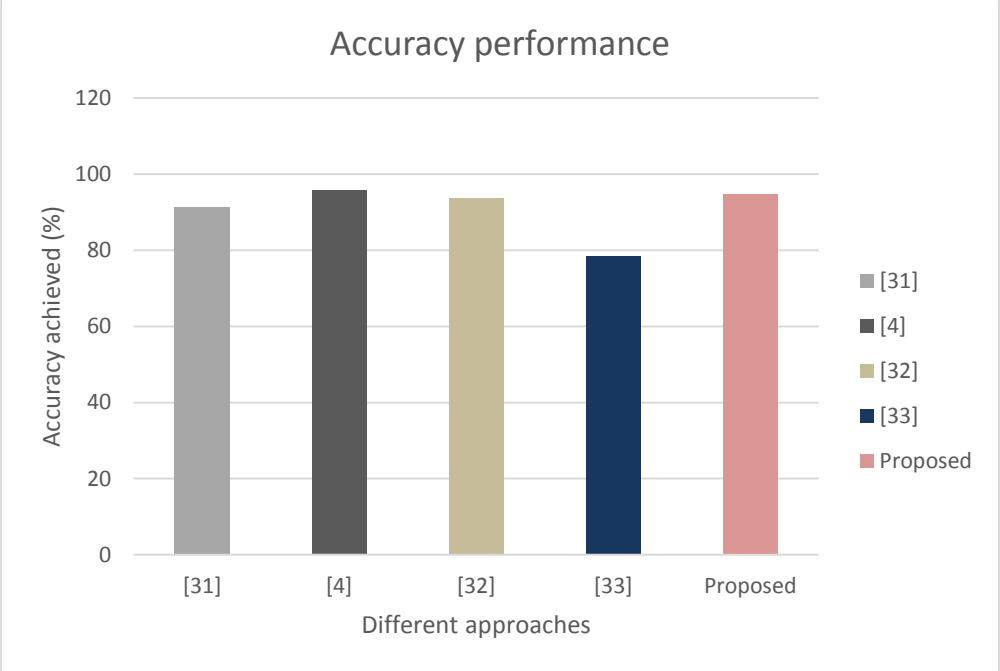

Figure 3. Accuracy performance of proposed POS Tagging and over state-of-art cross lingual learning technique

Table IV. Experimental comparsion over state-of-art monolingual technique

Experiment Id	**[7]**	**Proposed**
Language pair	Kannada	Kannada
Cross lingual support	No	Yes
Algorithm	n-gram CRF	HMM
Decoding algorithm used	-	Beam
Number of words	16,000 (80,000)	16000 (80,000)
Number of Tags used	36	25
Computation time (ms)	190000	60000

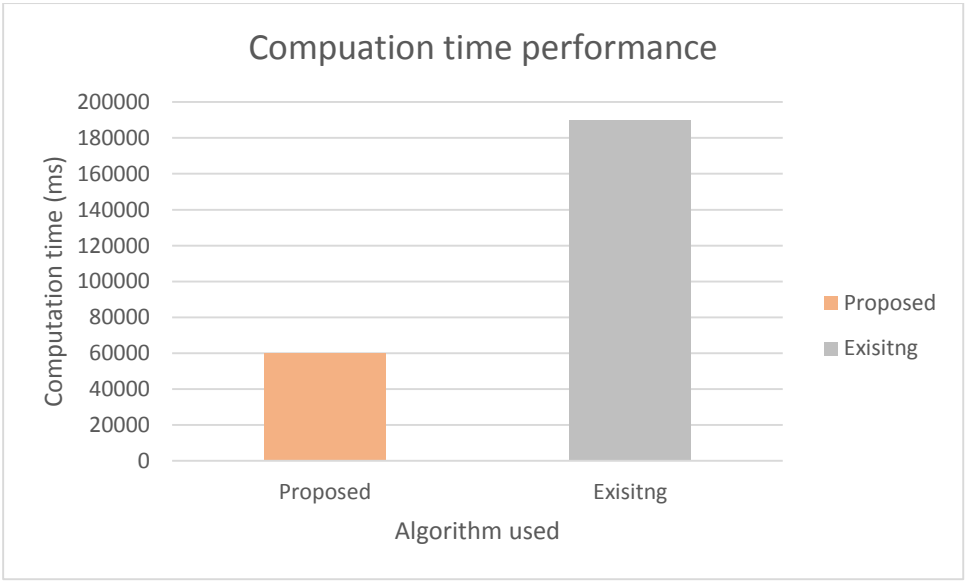

Figure 4. Computation time performance of proposed POS Tagging and over existing technique

CONCLUSION

We introduced a cross-lingual transfer learning model for POS tagging for resource-poor Dravidian languages (Kannada) using relatively resource-rich languages (Telugu). Experiment for Kannada are conducted using Telugu shows good results. It achieves better result than state-of-art monolingual and cross-lingual technique. We conducted extensive survey shows very less work for cross lingual learning for Indian languages is carried out. Our model achieves an accuracy of 94.62% for performing tagging on Kannada corpus which is significantly higher than state-of-art techniques. Our tagger is also fast irrespective of corpus size due to adoption of beam search. A 68.43% computation overhead reduction is achieved by proposed model over existing model. Our model achieves a good tradeoff between speedup and accuracy of tagging. The future work we consider working on building parallel dictionary for Kannada-Telugu pair using machine learning technique.

REFERENCES

[1] P. J. Antony and K. P. Soman, "Kernel based part of speech tagger for Kannada," 2010 International Conference on Machine Learning and Cybernetics, Qingdao, pp. 2139-2144, 2010.

[2] Jyoti Singh, Nisheeth Joshi, Iti Mathur, “Development of Marathi Part of Speech Tagger Using Statistical Approach”, In 2013 International Conference on Advances in Computing, Communications and Informatics, arXiv:1310.0575, 2013.

[3] Smruthi Mukund, Rohini Srihari, Erik Peterson, “An Information-Extraction System for Urdu---A Resource-Poor Language”, ACM Transactions on Asian Language Information Processing (TALIP), Volume 9 Issue 4, Article No. 15, December 2010.

[4] Jan Buys, Jan A. Botha, “Cross-Lingual Morphological Tagging for Low-Resource Languages”, Computation and Language, arXiv:1606.04279, 2016.

[5] Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, XIN-YU DAI, Jiajun Chen, “Word-Context Character Embeddings for Chinese Word Segmentation”, 2017 Conference on Empirical Methods in Natural Language Processing, 760–766, 2017.

[6] A. Bharati, R. Sangal, D. M. Sharma, and L. Bai. Anncorra: Annotating corpora guidelines for POS and chunk annotation for Indian languages. In Technical Report (TR-LTRC-31), LTRC, IIIT-Hyderabad, 2006.

[7] K. P. Pallavi, Anitha S. Pillai, “Kannpos-Kannada Parts of Speech Tagger Using Conditional Random Fields”, Emerging Research in Computing, Information, Communication and Applications. Springer, pp 479-491, 2016.

[8] D. Gunasekara, W. V. Welgama and A. R. Weerasinghe, "Hybrid Part of Speech tagger for Sinhala Language," 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer), Negombo, pp. 41-48, 2016.

[9] F. Rashel, A. Luthfi, A. Dinakaramani and R. Manurung, "Building an Indonesian rule-based part-of-speech tagger," 2014 International Conference on Asian Language Processing (IALP), Kuching, pp. 70-73, 2014.

[10] Z. Li, M. Zhang, W. Che, T. Liu and W. Chen, "Joint Optimization for Chinese POS Tagging and Dependency Parsing," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 1, pp. 274-286, Jan. 2014.

[11] D. Bahdanau, K. Cho, Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint, Sep 2014.

[12] Jiri Hana, Anna Feldman, and Chris Brew. A Resource-light Approach to Russian Morphology: Tagging Russian using Czech resources. In Proceedings of EMNLP, Barcelona, Spain, 2004.

[13] G. Nagaraju, N. Mangathayaru, B. Padmaja Rani, “Dependency Parser for Telugu Language”, Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, Article No. 138, 2016.

[14] Adam Kilgarriff and Gregory Grefenstette. Introduction to the special issue on the web as corpus. CL, 29(3):333–348, 2003.

[15] Adam Kilgarriff, Siva Reddy, Jan Pomik´alek, and Avinesh PVS. 2010. A corpus factory for many languages. In Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10), Valletta, Malta, may. European Language Resources Association (ELRA), 2010.

[16] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. In Selected papers from the sixth international conference on World Wide Web, pages 1157–1166, 1997.

[17] P. V. S. Avinesh and G. Karthik. Part-Of-Speech Tagging and Chunking using Conditional Random Fields and Transformation-Based Learning. In Proceedings of the IJCAI and the Workshop On Shallow Parsing for South Asian Languages (SPSAL), pages 21–24, 2007.

[18] Amaresh Datta. The Encyclopaedia Of Indian Literature, volume 2, 1998.

[19] Helmut Schmid and Florian Laws. Estimation of conditional probabilities with decision trees and an application to fine-grained pos tagging. In Proceedings of the 22nd International Conference on

Computational Linguistics - Volume 1, COLING '08, pages 777–784, Stroudsburg, PA, USA. Association for Computational Linguistics, 2008.

[20] Christer Samuelsson. Morphological tagging based entirely on Bayesian inference. In 9th Nordic Conference on Computational Linguistics NODALIDA-93, Stockholm University, Stockholm, Sweden, 1993.

[21] Jurafsky and Marting – Language modeling ,N-Grams and Corpora ; slides by Dan Jurafsky, 2013.

[22] Z. Li, M. Zhang, W. Che, T. Liu and W. Chen, "Joint Optimization for Chinese POS Tagging and Dependency Parsing," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 1, pp. 274-286, Jan. 2014.

[23] Y. Zhang and S. Clark, "A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing," in Proc. EMNLP '08, pp. 562–571, 2008.

[24] Y. Zhang and S. Clark, "Syntactic processing using the generalized perceptron and beam search," Comput. Linguist., vol. 37, no. 1, pp. 105–151, 2011.

[25] Bing: http://bing.com, Last access on October, 26, 2017.

[26] http://infohost.nmt.edu/~shipman/soft/deduper/deduper.pdf, last access on October 26, 2017.

[27] Shambhavi B R and Ramakanth Kumar P. Article: Kannada Part-Of-Speech Tagging with Probabilistic Classifiers. International Journal of Computer Applications 48(17):26-30, June 2012.

[28] Pallavi K.P., Pillai A.S. "Parts Of Speech (POS) Tagger for Kannada Using Conditional Random Fields (CRFs)" Conference: Conference: National Conference on Indian Language Computing (NCILC 2014).

[29] D. Gunasekara, W. V. Welgama and A. R. Weerasinghe, "Hybrid Part of Speech tagger for Sinhala Language," 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer), Negombo, 2016, pp. 41-48.

[30] P. J. Antony and K. P. Soman, "Kernel based part of speech tagger for Kannada," 2010 International Conference on Machine Learning and Cybernetics, Qingdao, pp. 2139-2144, 2010.

[31] Joo-Kyung Kim†, Young-Bum Kim‡, Ruhi Sarikaya‡, Eric Fosler-Lussier, "Cross-Lingual Transfer Learning for POS Tagging without Cross-Lingual Resources", Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2822–2828, 2017.

[32] Ryan Cotterell and Georg HeigoldK "Cross-lingual, Character-Level Neural Morphological Tagging", arXiv:1708.09157v1 [cs.CL] 30 Aug 2017.

[33] Long Duong, Paul Cook, Steven Bird and Pavel Pecina1, "Increasing the quality and quantity of source language data for unsupervised cross-lingual POS tagging", International Joint Conference on Natural Language Processing, pages 1243–1249, Nagoya, Japan, 14-18 October 2013.