Cross Language POS Taggers (and other Tools) for Indian Languages: An Experiment with Kannada using Telugu Resources

Siva Reddy
Lexical Computing Ltd, UK
siva@sketchengine.co.uk

Serge Sharoff
University of Leeds, UK
s.sharoff@leeds.ac.uk

Abstract

Indian languages are known to have a large speaker base, yet some of these languages have minimal or non-efficient linguistic resources. For example, Kannada is relatively resource-poor compared to Malayalam, Tamil and Telugu, which in-turn are relatively poor compared to Hindi. Many Indian language pairs exhibit high similarities in morphology and syntactic behaviour e.g. Kannada is highly similar to Telugu. In this paper, we show how to build a cross-language part-of-speech tagger for Kannada exploiting the resources of Telugu. We also build large corpora and a morphological analyser (including lemmatisation) for Kannada. Our experiments reveal that a cross-language taggers are as efficient as mono-lingual taggers. We aim to extend our work to other Indian languages. Our tools are efficient and significantly faster than the existing mono-lingual tools.

1 Introduction

Part-of-speech (POS) taggers are some of the basic tools for natural language processing in any language. For example, they are needed for terminology extraction using linguistic patterns or for selecting word lists in language teaching and lexicography. At the same time, many languages lack POS taggers. One reasons for this is the lack of other basic resources like corpora, lexicons or morphological analysers. With the advent of Web, collecting corpora is no longer a major problem (Kilgarriff et al., 2010). With technical advances in lexicography (Atkins and Rundell, 2008), building lexicons and morphological analysers is also possible to considerable extent.

The other reason for the lack of POS taggers is partly due the lack of researchers working on a particular language. Due to this, some languages do not have any annotated data to build efficient taggers.

Cross-language research mainly aims to build tools for a resource-poor language (target language) using the resources of a resource-rich language (source language). If the target language is typologically related to the source one, it is possible to rely on the resource rich language.

In this work, we aim to find if cross language tools for Indian languages are any efficient as compared to existing mono-lingual tools. As a use case, we experiment with the resource-poor language Kannada, by building various cross-language POS taggers, using the resources of its typologically-related and relatively resource-rich language Telugu. Our POS taggers can also be used as a morphological analyser since our POS tags include morphological information. We also build a lemmatiser for Kannada which uses POS tag information to choose a relevant lemma from the set of plausible lemmas.

2 Related Work

There are several methods for building POS taggers for a target language using source language resources. Some researchers (Yarowsky et al., 2001; Yarowsky and Ngai, 2001; Das and Petrov, 2011) 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. Other methods which make use of parallel corpora are (Snyder et al., 2008; Naseem et al., 2009). These approaches are based on hierarchical Bayesian models and Markov Chain Monte Carlo

sampling techniques. 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.

Hana et al. (2004) and Feldman et al. (2006) propose a method for developing a POS tagger for a target language using the resources of another typologically related language. Our method is motivated from them, but with the focus on resources available for Indian languages.

2.1 Hana et al. (2004)

Hana et al. aim to develop a tagger for Russian from Czech using TnT (Brants, 2000), a second-order Markov model. Though the languages Czech and Russian are free-word order, they argue that TnT is as efficient as other models.

TnT tagger is based on two probabilities - the transition and emission probabilities. The tag sequence of a given word sequence is selected by calculating

$$\underset{t_1 \ldots t_n}{\operatorname{argmax}} \left[\prod_{i=1}^{n} P(t_i|t_{i-1}, t_{i-2}) P(w_i|t_i) \right] \quad (1)$$

where $w_i \ldots w_n$ is the word sequence and $t_1 \ldots t_n$ are their corresponding POS tags.

Transition probabilities, $P(t_i|t_{i-1}, t_{i-2})$, describe the conditional probability of a tag given the tags of previous words. Based on the intuition that transition probabilities across typologically related languages remain the same, Hana et al. treat the transition probabilities of Russian to be the same as Czech.

Emission probabilities, $P(w_i|t_i)$, describe the conditional probability of a word given a tag. It is not straightforward to estimate emission probabilities from a cross-language. Instead, Hana et al. develop a light paradigm-based (a set of rules) lexicon for Russian which emits all the possible tags for a given word form. The distribution of all the tags of a word is treated to be uniform. Using this assumption, surrogate emission probabilities of Russian are estimated without using Czech.

The accuracy of the cross-pos tagger, i.e. the tagger of Russian built using Czech, is found to be encouraging.

2.2 Existing Tools for Kannada

There exists literature on Kannada morphological analysers (Vikram and Urs, 2007; Antony et al., 2010; Shambhavi et al., 2011) and POS taggers (Antony and Soman, 2010) but none of them have any publicly downloadable resources. Murthy (2000) gives an overview of existing resources for Kannada and points out that most of these exist without public access. We are interested only in the work whose tools are publicly available for download.

We found only one downloadable POS tagger for Kannada developed by the Indian Language Machine Translation (ILMT) consortium[1]. The consortium publicly released tools for 9 Indian languages including Kannada and Telugu. The available tools are transliterators, morphological analysers, POS taggers and shallow parsers.

The POS taggers from the ILMT consortium are mono-lingual POS taggers i.e. trained using the target language resources itself. These were developed by Avinesh and Karthik (2007) by training a conditional random fields (CRF) model on the training data provided by the participating institutions in the consortium. In the public evaluation of POS taggers for Indian languages (Bharati and Mannem, 2007), the tagger (Avinesh and Karthik, 2007) was ranked best among all the existing taggers.

Indian languages are morphologically rich with Dravidian languages posing extra challenge because of their agglutinative nature. Avinesh and Karthik (2007) noted that morphological information play an important role in Indian language POS tagging. Their CRF model is trained on all the important morphological features to predict the output tag for a word in a given context. The pipeline of (Avinesh and Karthik, 2007) can be described as below

1. Tokenise the Unicode input
2. Transliterate the tokenised input to ASCII format.
3. Run the morph analyser to get all the morphological sets possible
4. Extract relevant morphological features used by the CRF model
5. Given a word, based on the morphological features of its context and itself, the CRF

[1]Tools for 9 Indian languages `http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php`

Field	Description	Number of Tags	Tags
	Full Tag	311	NN.n.f.pl.3.d, VM.v.n.sg.3., . . .
1	Main POS Tag	25	CC, JJ, NN, VM, . . .
2	Coarse POS Category	9	adj, n, num, unk . . .
3	Gender	6	any, f, m, n, punc, null
4	Number	4	any, pl, sg, null
5	Person	5	1, 2, 3, any, null
6	Case	3	d, o, null

Table 1: Fields in each tag and its corresponding statistics. *null* denotes empty value, e.g. in the tag *VM.v.n..3.*, *number* and *case* fields are *null*

model annotate the word with a relevant POS tag
6. Transliterate the ASCII output to Unicode

The major drawback with this tagging model is that it relies on a pipeline and if something breaks in the pipeline, the POS tagger doesn't work. We found that the tagger annotates only 78% of the input sentences. The tagger is found to be too slow to scale for large annotation tasks.

We aim to remove this pipeline, yet build an efficient tagger which also performs morphological analysis at the same time.

2.3 Kannada and Telugu Background

Kannada and Telugu are spoken by 35 and 75 million people respectively[2]. Majority of the existing research in Indian languages focused on few languages like Hindi, Marathi, Bengali, Telugu and Tamil, as a result of which other languages like Kannada, Malayalam are relatively resource-poor.

Telugu is known to be highly influenced by Kannada, making the languages slightly mutually intelligible (Datta, 1998, pg. 1690). Until 13^{th} century both the languages have same script. In the later years, the script has changed but still close similarities can be observed. Both the scripts belong to the same script family.

The similarities between Kannada and Telugu, and the relative resource abundance in Telugu, motivates us to develop a cross language POS tagger for Kannada using Telugu.

3 Our Tagset

All the Indian languages have similarities in morphological properties and syntactic behaviour. The only main difference is the agglutinative behaviour of Dravidian languages. Observing these similarities and differences in Indian languages, Bharati et al. (2006) proposed a common POS tagset for all Indian languages. Avinesh and Karthik (2007) use this tagset.

We encode morphological information to the above tagset creating a *fine-grained POS tagset* similar to the work of (Schmid and Laws, 2008) for German, which is morphologically rich like Kannada. Each tag consists of 6 fields. Table 1 describe each field and its statistics. For example, our tag **NN.n.m.sg.3.o** represents the main POS tag 'NN' for *common noun* as defined by (Bharati et al., 2006), 'n' for coarse grained category *noun*, 'm' for *masculine* gender, 'sg' for *singular* number, '3' for 3^{rd} *person*, 'o' for *oblique* case. For more guidelines on morphological labels, please refer to (Bharati et al., 2007).

Since our POS tag encodes morphological information in itself, our tagger could also be used as a morphological analyser. A sample sentence POS tagged by our tagger is displayed in Figure 1.

4 Our Method

We aim to build a Hidden-Markov model (HMM) based Kannada POS tagger described by the Equation 1. We use TnT (Brants, 2000), a popular implementation of the second-order Markov model for POS tagging. We construct the TnT model by estimating transition and emission probabilities of Kannada using the cross-language Telugu. Since our tagset has both POS and morphological information encoded in it, the HMM model has an advantage of using morphological information to predict the main POS tag, and the inverse, where main POS tag helps to predict the morphological information. Briefly, the steps involved in our method are

1. Download large corpora of Kannada and Telugu

[2]Source: Wikipedia

Word	POS Tag	Lemma.Suffix
ಕತೆಯ	NN.n.n.sg..o	ಕತೆ.ಅ
ಪ್ರಕಾರ	NN.n.n.sg..d	ಪ್ರಕಾರ.ಂ
ಗೆಳೆಯರೊಂದಿಗಿನ	NN.unk....	ಗೆಳೆಯರೊಂದಿಗಿನ.
ಆಟದಲ್ಲಿ	NN.n.n.sg..o	ಆಟ.ಅಲ್ಲಿ
ರಾಜನಾಗಿದ್ದ	VM.unk....	ರಾಜನಾಗಿದ್ದ.
ಚಂದ್ರಗುಪ್ತನು	NNP.unk....	ಚಂದ್ರಗುಪ್ತನು.
ಅಪರಾಧಿಯ	NN.n.m.sg.3.o	ಅಪರಾಧಿ.ಅ
ಪಾತ್ರ	NN.n.n.sg..d	ಪಾತ್ರ.ಂ
ವಹಿಸಿದ್ದ	VM.v.any.any.any.	ವಹಿಸು.ಇದ್ದ
ಇನ್ನೊಬ್ಬ	QC.unk....	ಇನ್ನೊಬ್ಬ.
ಹುಡುಗನ	NN.n.m.sg.3.o	ಹುಡುಗ.ಅ
ವಿಚಾರಣೆಯನ್ನು	NN.n.n.sg..o	ವಿಚಾರಣೆ.ಅನ್ನು
ಮಾಡಿ	VM.v..pl.2.	ಮಾಡು.ಂ
ಶಿಕ್ಷೆ	NN.n.n.sg..d	ಶಿಕ್ಷೆ.ಂ
ವಿಧಿಸುತ್ತಿದ್ದನು	VM.v.m.sg.3.	ವಿಧಿಸು.ಉತ್ತಿದ್
.	SYM.punc....	..

Figure 1: A Sample POS Tagging and Lemmatisation for a Kannada Sentence

2. Determine the transition probabilities of Telugu by training TnT on the machine annotated corpora of Telugu. Since Telugu and Kannada are typologically related, we assume the transition probabilities of Kannada to be the same as of Telugu
3. Estimate the emission probabilities of Kannada from machine annotated Telugu corpus or machine annotate Kannada corpus
4. Use the probabilities from the step 2 and 3 to build a POS tagger for Kannada

4.1 Step1: Kannada and Telugu Corpus Creation

Corpus collection once used to be long, slow and expensive. But with the advent of the Web and the success of Web-as-Corpus notion (Kilgarriff and Grefenstette, 2003), corpus collection can be highly automated, and thereby fast and inexpensive.

We have used *Corpus Factory* method (Kilgarriff et al., 2010) to collect Kannada and Telugu corpora from the Web. The method is described in the following steps.

Frequency List: Corpus Factory method requires a frequency list of the language of interest to start corpus collection. The frequency list of the language is built from its Wikipedia dump[3]. The dump is processed to remove all the Wiki and HTML markup to extract raw corpus, the Wiki corpus. The frequency list is then built from the tokenised Wiki corpus.

Seed Word Collection: We treat the top 1000 words of the frequency list as the *high-frequency words* of the language and the next 5000 as the mid-frequency ones which we shall use as our *seed words*.

Query Generation: 30,000 random queries of 2-word size are generated such that no query is identical nor its permutations.

URL Collection: Each query is sent to Bing[4] search engine and the pages corresponding to the hits are downloaded. These pages are converted to UTF-8 encoding.

Filtering Above pages are cleaned to remove boiler-plate text (i.e. html and irrelevant blocks like ads) extracting the plain text. Some of these pages are found to be in foreign languages and some of them are found to be spam. We applied a simple language modelling based filter to remove these pages. The filter validates only the pages in

[3]Wikipedia Dumps: `http://dumps.wikimedia.org`

[4]Bing: `http://bing.com`

which the ratio of non-frequent words to the high-frequent words is maintained. If a page doesn't meet this criteria, we discard it.

Near-Duplicate Removal: The above filter isn't sufficient to discard the pages which are duplicates. In-order to detect them, we used Broder et al. (1997) near-duplicate detection algorithm, and store only one page among the duplicates.

Finally we collected cleaned corpora of 16 million words for Kannada and 4.6 million words for Telugu[5].

4.2 Step 2: Estimating Kannada Transition Probabilities

Transition probabilities represent the probability of transition to a state from the previous states. Here each state represents a tag and hence $P(t_i|t_{i-1}, t_{i-2})$. We estimate transition probabilities in two different ways.

4.2.1 From the source language

Across typologically related languages, it is likely that transition probabilities among tags are the same. We assume the transition probabilities of Telugu to be approximately equal to the transition probabilities of Kannada.

One can estimate the transition probabilities of a language from its manually annotated corpora. Since we do not have the manually annotated Telugu corpora publicly available, we have used (Avinesh and Karthik, 2007) to tag the Telugu corpus downloaded in Step 1. This tagged corpus captures an approximation of the true transition probabilities in the manually annotated corpora.

The tagged corpus is converted to the format in Figure 1 and then using TnT we estimate transition probabilities.

4.2.2 From the target language

Apart from using Telugu transition probabilities, we also experimented with the existing Kannada POS tagger. We annotated the Kannada corpus collected in Step 1 using the existing tagger. We then estimated the transition probabilities from the machine annotated Kannada corpus. Note that if Kannada POS tagger is used for estimating transition probabilities, our tagger can no longer be called a cross-language tagger, and is mono-lingual. This tagger is used to compare the performance of cross-lingual and mono-lingual taggers.

[5]Telugu is collected two years back and Kannada very recently and so are the differences in sizes.

Since we learn the transition probabilities of the fine-grained POS tags from a large corpora, this helps in building a robust and efficient tagger compared to the existing mono-lingual tagger. Robust because we would be able to predict POS and morphological information for unseen words, and efficient because the morphological information helps in better POS prediction and vice versa.

4.3 Step 3: Estimating Kannada Emission Probabilities

Emission probabilities represent the probabilities of an emission (output) of a given state. Here state corresponds to tag and emission to a word and hence $P(w_i|t_i)$. We tried various ways of estimating emission probabilities of Kannada.

4.3.1 Approximate string matching

It is not easy to estimate emission probabilities of a language from a cross language without the help of either parallel corpora or a bilingual dictionary or a translation system. Since the languages, Kannada and Telugu, are slightly mutually intelligible (Datta, 1998, pg. 1690), we aimed to exploit lexical similarities between Kannada and Telugu to the extent possible.

Firstly, a Telugu lexicon is built by training TnT on the machine annotated Telugu corpora (Step 1). The lexicon has the information of each Telugu word and its corresponding POS tags along with their frequencies. Then, a word list for Kannada is built from the Kannada corpus. For a every Kannada word, the most probable similar Telugu word is determined using approximate string matching [6]. To measure similarity, we transliterated both Kannada and Telugu words to a common ASCII encoding. For example, the most similar Telugu words of the Kannada word *xAswAnu* are ('xAswAn', 0.545), ('xAswAru', 0.5), ('rAswAnu', 0.5), ('xAswAdu', 0.5) and the most similar Telugu words of the Kannada word *viBAgavu* are ('viBAgamu', 0.539), ('viBAga', 0.5), ('viBAgalanu', 0.467), ('viBAgamulu', 0.467).

We assume that for a Kannada word, its tags and their frequencies are equal to the most similar Telugu word. Based on this assumption, we build a lexicon for Kannada with each word having its plausible tags and frequencies derived from Telugu. This lexicon is used for estimating transition probabilities.

[6]We used Python n-gram package for approximate string matching: `http://packages.python.org/ngram/`

4.3.2 Source tags and target morphology

For each morphological set from the machine annotated Telugu corpora, we determine all its plausible fine-grained POS tags. For example, morphological set **n.n.sg..o** is associated with all the tags which satisfy the regular expression ***.n.n.sg..o**. Then for every word in Kannada, based on its morphology determined by the morphological analyser, we assign all the tags applicable, as learned from Telugu uniformly. The drawback of this approach is that the search space is large.

4.3.3 Target tags with uniform distribution

Instead of estimating emission probabilities from the cross language, we learn the plausible fine-grained tags of a given Kannada word from the machine annotated Kannada corpora (Step 1) and assume uniform distribution over all its tags. Though we learn the tags using the existing POS tagger, we do not use the information about tag frequencies, and hence we are not using the emission probabilities of the existing tagger. The existing tagger is just used to build a lexicon for Kannada.

Since we run the tagger on a large Kannada corpus, our lexicon contains most of the Kannada word forms and their corresponding POS and morphological information. This lexicon helps in removing the pipeline of (Avinesh and Karthik, 2007), thus building a high-speed tagger. Even, if some words are absent in the lexicon, TnT is well known to predict tags for unseen words based on the transition probabilities.

The advantage of this method over the previous is that the search space is drastically reduced.

4.3.4 Target emission probabilities

In this method, we learn the Kannada emission probabilities directly from the machine annotated Kannada corpora, i.e. we use the emission probabilities of the existing tagger. This helps us in estimating the upper-bound performance of the cross-lingual tagger when the transition probabilities are taken from Telugu.

Also, it helps in estimating the upper-bound performance of mono-lingual tagger when the transition probabilities are directly taken from Kannada. Our mono-lingual tagger will be robust, fast and as accurate as the existing mono-lingual tagger.

4.4 Step4: Final Tagger

We experimented with various TnT tagging models by selecting transition and emission probabilities from the Steps 2 and 3. Though one may question the performance of TnT for free-word order languages like Kannada, Hana et al. (2004) found that TnT models are as good as other models for free-word order languages. Additionally, Schmid and Laws (2008) observed that TnT models are also good at learning fined-grained transition probabilities. In our evaluation, we also found that our TnT models are competitive to the existing CRF model of (Avinesh and Karthik, 2007).

Apart from building POS tagging models, we also learned the associations of each word with its lemma and suffix given a POS tag, from the machine annotated Kannada corpus. For example, Kannada word *aramaneVgalYannu* is associated with lemma *aramaneV* and suffix *annu* when occurred with the tag NN.n.n.pl..o and similarly word *aramaneVgeV* is associated with lemma *aramaneV* and suffix *igeV* when occurred with the tag NN.n.n.sg..o.

An example sentence tagged by our models along with the lemmatisation is displayed in Figure 1.

5 Evaluation Results

We evaluated all our models on the manually annotated Kannada corpora developed by the ILMT consortium[7]. The corpus consists of 201,373 words and it is tagged with Bharati et al. (2006) tagset which forms the first field of our fine-grained POS tagset. Since we did not have manually annotated data for morphology, we evaluated only on the first field of our tags. For example, in the tag *NST.n.n.pl..o*, we evaluate only for *NST*.

Table 2 displays the results for various tagging models. Note that all our models are TnT models whereas (Avinesh and Karthik, 2007) is a CRF model.

Model 1 uses the transition probabilities of Telugu (section 4.2.1) and emission probabilities estimated from Telugu using approximate string matching (section 4.3.1). This model achieves 50% accuracy without using almost any resources of the target language. This is encouraging especially for languages which do not have any re-

[7]This corpus is not publicly available and is licensed. We did not use it for any of our training purposes except for the evaluation

Model	Transition Prob	Emission Prob	Precision	Recall	F-measure
		Cross-Language POS Tagger			
1	From the source language	Approximate string matching	56.88	56.88	56.88
2	From the source language	Source tags and target morphology	28.65	28.65	28.65
3	From the source language	Target tags with uniform distribution	75.10	75.10	75.10
4	From the source language	Target emission probabilities	77.63	77.63	77.63
		Mono-Lingual POS Tagger			
5	From the target language	Target emission probabilities	77.66	77.66	77.66
6	(Avinesh and Karthik, 2007)		78.64	61.48	69.01

Table 2: Evaluation results of various tagging models

sources.

Model 2 uses the transition probabilities of Telugu (section 4.2.1) and the emission probabilities estimated by mapping Telugu tags to the Kannada morphology (section 4.3.2). The performance is poor due to explosion in search space of the plausible tags. We optimise the search space using a Kannada lexicon in Model 3.

Model 3 uses the transition probabilities of Telugu (section 4.2.1) and emission probabilities estimated from machine-built Kannada lexicon (section 4.3.3). The performance is competitive with the mono-lingual taggers Models 5 and 6. The tagger has better F-measure than (Avinesh and Karthik, 2007). This model reveals that transition probabilities apply across typologically related Indian languages. To build an efficient cross-lingual tagger, it is good-enough to use cross-language transitions along with the target lexicon i.e. the list of all the tags plausible for a given target word.

Model 4 uses the transition probabilities of Telugu (section 4.2.1) and emission probabilities of Kannada estimated from the existing Kannada tagger (section 4.3.4). This gives us an idea of the upper-bound performance of cross-language POS taggers when source transition probabilities are used. The performance is almost equal to the mono-lingual tagger Model 5, showing that transition probabilities across Kannada and Telugu are almost same. We could build cross-language POS taggers as efficient as mono-lingual taggers conditioned that we have a good target lexicon.

Model 5 is a mono-lingual tagger which uses target transition and emission probabilities estimated from the existing tagger (section 4.2.2 and 4.3.4). The performance is highly competitive with better F-measure than (Avinesh and Karthik, 2007). This shows that a HMM-based tagger is as efficient as a CRF model (or any other model). While to tag 16 million words of Kannada corpora using (Avinesh and Karthik, 2007) took 5 days on a Quadcore processor @ 2.3 GHz each core, it hardly took few minutes by TnT model with better recall. We also aim to develop robust, fast and efficient mono-lingual taggers to Indian languages which already have POS taggers.

Table 3 displays the tagwise results of our cross-language tagger Model 3, our mono-lingual tagger Model 5 and the existing mono-lingual tagger Model 6.

6 Conclusions

This is an attempt to build POS taggers and other tools for resource-poor Indian languages using relatively resource-rich languages. Our experimental results for Kannada using Telugu are highly encouraging towards building cross-language tools. Cross-language POS taggers are found to be as accurate as the mono-lingual POS taggers.

Future directions include building cross language tools for other resource-poor Indian language, such as Malayalam using Tamil, Marathi using Hindi, Nepali using Hindi, etc. For Indian languages which already have tools, we aim to build robust, fast and efficient tools using the existing tools.

Finally, all the tools developed in this work are available for download[8]. The corpora (tagged) developed for this work is accessible through Sketch Engine[9] or Intellitext[10] interface.

Acknowledgements

This work has been supported by AHRC DEDEFI grant AH/H037306/1. Thanks to anonymous reviwers for thier useful feedback.

[8]The tools developed in this work can be downloaded from `http://sivareddy.in/downloads` or `http://corpus.leeds.ac.uk/serge/`

[9]Sketch Engine `http://sketchengine.co.uk`

[10]Intellitext `http://corpus.leeds.ac.uk/it/`

Tag	Freq	Prec	Recall	F1	Prec	Recall	F1	Prec	Recall	F1
		Model 3			Model 5			Model 6		
NN	81289	74.32	84.89	79.25	81.58	80.79	81.19	84.91	62.59	72.06
VM	33421	84.56	88.21	86.35	83.94	89.39	86.58	86.79	71.78	78.57
SYM	30835	92.26	95.51	93.86	95.57	96.11	95.84	95.64	73.99	83.43
JJ	13429	54.92	27.59	36.73	55.54	39.70	46.30	56.38	32.76	41.44
PRP	9102	60.02	33.14	42.70	59.07	56.01	57.50	60.69	46.07	52.38
QC	7699	90.70	73.45	81.17	90.55	93.52	92.01	88.52	70.40	78.43
NNP	7221	43.66	45.41	44.52	60.87	61.82	61.34	62.20	61.72	61.96
CC	4003	87.11	92.03	89.50	88.62	94.33	91.38	88.69	75.39	81.50
RB	3957	27.03	26.26	26.64	33.48	37.30	35.29	34.31	29.52	31.73
NST	2139	49.26	62.51	55.10	38.72	79.34	52.04	40.27	67.27	50.39
QF	1385	67.17	80.36	73.18	54.95	80.51	65.32	58.18	70.61	63.80
NEG	889	68.00	3.82	7.24	89.93	42.18	57.43	86.50	35.32	50.16
QO	622	54.66	20.74	30.07	45.43	28.78	35.24	54.00	21.70	30.96
WQ	599	70.25	46.91	56.26	80.17	80.30	80.23	81.73	55.26	65.94
PSP	374	7.92	2.14	3.37	-	-	-	26.28	71.39	38.42
INTF	23	5.32	43.48	9.48	5.08	60.00	9.38	1.06	17.39	2.00
INJ	3	5.13	66.67	9.52	1.67	33.33	3.17	2.70	33.33	5.00
Overall	201,373	75.10	75.10	75.10	77.66	77.66	77.66	78.64	61.48	69.01

Table 3: Tag wise results of Models 3, 5 and 6 described in Table 2

References

P.J. Antony and K.P. Soman. 2010. Kernel based part of speech tagger for kannada. In *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, volume 4, pages 2139 –2144, july.

P.J. Antony, M Anand Kumar, and K.P. Soman. 2010. Paradigm based morphological analyzer for kannada language using machine learning approach. *Advances in Computational Sciences and Technology (ACST)*, 3(4).

Sue B. T. Atkins and Michael Rundell. 2008. *The Oxford Guide to Practical Lexicography*. Oxford University Press, Oxford.

P. V. S. Avinesh and G. Karthik. 2007. 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.

Akshar Bharati and Prashanth R. Mannem. 2007. Introduction to shallow parsing contest on south asian languages. In *Proceedings of the IJCAI and the Workshop On Shallow Parsing for South Asian Languages (SPSAL)*, pages pages 1–8.

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

A. Bharati, R. Sangal, and D.M. Sharma. 2007. Ssf: Shakti standard format guide. Technical Report TR-LTRC-33, Language Technologies Research Centre, IIIT-Hyderabad, India.

Thorsten Brants. 2000. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of ACL 2011*.

Amaresh Datta. 1998. *The Encyclopaedia Of Indian Literature*, volume 2.

Anna Feldman, Jirka Hana, and Chris Brew. 2006. A cross-language approach to rapid creation of new morpho-syntactically annotated resources. In *Proceedings of LREC*, pages 549–554.

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

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

Adam Kilgarriff, Siva Reddy, Jan Pomikálek, 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).

Kavi Narayana Murthy. 2000. Computer processing of kannada language. Technical report, Computer and Kannada Development, Kannada University, Hampi.

Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. 2009. Multilingual part-of-speech tagging: Two unsupervised approaches. *J. Artif. Intell. Res. (JAIR)*, 36:341–385.

Helmut Schmid and Florian Laws. 2008. 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.

B. R Shambhavi, P Ramakanth Kumar, K Srividya, B J Jyothi, Spoorti Kundargi, and G Varsha Shastri. 2011. Kannada morphological analyser and generator using trie. *IJCSNS International Journal of Computer Science and Network Security*, 11(1), January.

Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. 2008. Unsupervised multilingual learning for pos tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1041–1050, Stroudsburg, PA, USA. Association for Computational Linguistics.

T. N. Vikram and Shalini R. Urs. 2007. Development of prototype morphological analyzer for the south indian language of kannada. In *Proceedings of the 10th international conference on Asian digital libraries: looking back 10 years and forging new frontiers*, ICADL'07, pages 109–116, Berlin, Heidelberg. Springer-Verlag.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the first international conference on Human language technology research*, HLT '01, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.