Developing Part-of-Speech Tagger for a Resource Poor Language : Sindhi

Raveesh Motlani*, Harsh Lalwani†, Manish Shrivastava*, Dipti Misra Sharma*

*Language Technology Research Centre
International Institute of Information Technology-Hyderabad
Telangana-500032, India
raveesh.motlani@research.iiit.ac.in, {m.shrivastava, dipti}@iiit.ac.in

†Department of Computer Science
Jabalpur Engineering College, Jabalpur
Madhya Pradesh-482011, India
harsh.lalwani@jecjabalpur.ac.in

Abstract

Sindhi is an Indo-Aryan language spoken by more than 58 million speakers around the world. It is currently a resource poor language which is harmed by the literature being written in multiple scripts. Though the language is widely spoken, primarily, across two countries, the written form is not standardized. In this paper, we seek to develop resources for basic language processing for Sindhi language, in one of its preferred scripts (Devanagari), because a language that seeks to survive in the modern information society requires language technology products. This paper presents our work on building a stochastic Part-of-Speech tagger for Sindhi-Devanagari using conditional random fields with linguistically motivated features. The paper also discusses the steps taken to construct a part-of-speech annotated corpus for Sindhi in Devanagari script. We have also explained in detail the features that were used for training the tagger, which resulted in a part of speech tagger nearing 92% average accuracy.

1. Introduction

Sindhi is an Indo-Aryan language spoken by about 53 million people in Pakistan and about 5.8 million people in India. Sindhi is also one of the 22 official languages in India[1]. Despite all these statistics showing how widely spoken Sindhi is, it is still a computationally resource poor language.

Historically, Sindhi has been written using many writing systems such as Landa, Waranki, Khudawadi, Gurmukhi, Perso-Arabic and Devanagari. The literature (Daswani, 1979) says that during the colonial rule, the British regime faced a problem in recognizing the major prevalent script out of many others for Sindhi and after prolonged deliberation they chose Perso-Arabic as the official script for Sindhi in 1853.

Later, when the partition of India and Pakistan took place in 1947, Sindh (the region where majority of Sindhi speaking people resided) became a part of Pakistan. A lot of Sindhi speaking people migrated to India and spread across the country. When the question of declaring a standard script for Sindhi in India came up, groups supporting either Perso-Arabic or Devanagari stood up. Initially, the Indian Government declared Devanagari as standard script but owing to protests, both scripts were accepted. However, the Perso-Arabic script remained standard in Pakistan.

Sindhi speaking population does not have a geographical state in India. Hence, despite being scheduled language, it is not used as an official language anywhere and therefore, does not have much literature. On the other hand, there is Sind in Pakistan and people have been contributing to the language in various ways. For instance, there exists a Sindhi Wikipedia, various newspapers and blogs in Perso-Arabic, published from Pakistan. In contrast, there is very little text in Sindhi in Devanagari (henceforth Sindhi-Devanagari) on the web.

Though both the scripts are used in India, it is important that we leverage Devanagari, since it is more prevalent in India and is shared with other Indian Languages, specifically, Hindi. Hence, we are trying to build resources and tools for Sindhi-Devanagari. Some of the resources are POS annotated corpus and POS tagger. POS tagging is a very important preprocessing task for language processing activities. Our aim in this paper is to develop a stochastic POS tagger for Sindhi-Devanagari.

This paper is organized in the following manner. We introduced our problem in Section 1. and the related work done is described in Section 2. In Section 3, we talk about our corpus and the annotation scheme. Section 4 describes the experimental setup, the algorithm of Conditional Random Fields and our evaluation metrics. In Section 5. we have discussed the major experiments conducted, their results and analysis. In the end, we have given our conclusion and described the future work in Section 6.

2. Related Work

Part-of-Speech tagging is the process of assigning a part-of-speech label to each word in a sentence, based

[1] The Constitution of India. page 330, EIGHTH SCHEDULE, Articles 344 (1) and 351. Languages.

on both, its definition as well as its context. Traditionally, many different approaches have been used for part of speech tagging. One such linguistically motivated approach is that of a rule based part-of-speech tagger. On the other hand, there are machine learning algorithms such as Decision Trees (Black et al., 1992), HMMs (Cutting et al., 1992; Brants, 2000), MEMMs, CRFs (Lafferty et al., 2001), Maximum Entropy (Ratnaparkhi, 1996), etc. Sometimes, people also combine both to form a Hybrid tagger, such as CLAWS (Garside and Smith, 1997). The algorithm that we have used (Conditional Random Fields) is a statistical one.

Conditional Random Fields have been used for creating taggers for a long time now. They were first used for POS tagging experiments by (Lafferty et al., 2001). Then, they were also used for the task of shallow parsing by (Sha and Pereira, 2003) where CRFs were applied for NP chunking of English on WSJ corpus and reported an accuracy of 94.38%. Later, they were used for POS tagging several Indian languages as well. CRFs were used for POS tagging on Hindi by (Shrivastava et al., 2006), on Bengali by (Ekbal et al., 2007), on Manipuri by (Nongmeikapam and Bandyopadhyay, 2012), on Gujarati by (Patel and Gali, 2008), on Kannada by (Shambhavi and Kumar P, 2012) and on various Indian languages. This is because Indian languages are morphologically rich and CRFs give the freedom to incorporate this and other linguistic properties of a language while training a POS tagger.

When it comes to Sindhi language, some work has been done using Perso-Arabic as the preferred script. A rule based POS tagger was developed by (Mahar and Memon, 2010). They developed a lexicon of 26,355 entries and a tagset containing 67 tags. Using both these resources along with about 186 disambiguation rules, their Sindhi POS tagger reported an accuracy of 96.28% . They have also contributed towards other aspects of natural language processing such as text segmentation, language modeling, etc. (Rahman and Bhatti, 2010) have worked on capturing Sindhi noun inflections through Finite State Machines. Unfortunately, these works could not be replicated for Sindhi-Devanagari because these resources are not available publicly. Also, it has been over 69 years since the partition of India and Pakistan took place and due to influence of Urdu and Punjabi in Pakistan, the Sindhi vocabulary has grown and diverged from the one spoken in India. Sindhi used in India has been influenced by Hindi, Gujarati, Punjabi and other Indian languages. Therefore, the lexicon and resources developed for Sindhi in Pakistan cannot be used directly.

3. Sindhi Corpus

The amount of raw text available on the web for Sindhi-Devanagari is very small. The problem was further compounded as many publishers on the web have not yet moved to Unicode standards. We contacted various publishers and news agencies to source raw data which could be annotated with POS tags. Eventually, most of the data collected was manually typed with Unicode encoding for Devanagari and annotated with POS tags. This manual process allowed us to correct words with incorrect encoding and grammatically incorrect sentences. We picked up stories, articles, news, general conversation from the web to create a corpus of 280k words.

3.1. Corpus Annotation

To start with this task, we did not have an annotation scheme and tagset defined for Sindhi. So, we adapted an existing and standardized BIS (Bureau of Indian Standards) tagset [2] by filtering out some tags which were not required or not applicable to the language. The BIS Tagset had been designed under the banner of Bureau of Indian Standards. This POS schema is based on W3C XML Internalization best practices. The BIS Tagset contains the features of a hierarchical tagset. However, it has tags for only first two tiers of linguistic information (POS and their subtypes) and excludes information from tier three onwards as these can be provided by morph analyzers and parsers. The BIS Tagset is comprehensive and is designed to be extensible to any Indian Language tagset. An example of a tagged sentence is given in Figure 1.

4. Experimental Setup

The corpus developed for Sindhi POS tagging was sourced from various web sources, articles and books. Any sentences containing non-Devanagari words were discarded. The annotation was done using an adaptation of BIS Tagset for Sindhi. So far, this had resulted in an annotated corpus of 37162 words. Ten fold cross-validation was employed by taking 90% data for training and 10% for testing for each fold. The linguistically motivated features which were used with CRF, are described below. Since not much work has gone into NLP tools for Sindhi, we do not have fundamental tools like morphological analyzers for the language. Thus, we have used features that try to emulate stemmers or morphological analyzers. This is essential as Sindhi is a morphologically rich language and CRF features have to be accordingly defined for better performance, as we shall see later.

4.1. Conditional Random Fields

Conditional Random Fields (CRFs) (Lafferty et al., 2001) is a form of undirected graphical model that defines a single log-linear distribution over label sequences given a particular observation sequence. CRFs calculate conditional probabilities of values on the output nodes, given values of input node in an undirected graph. So, the conditional probability of a state sequence $S =< s_1, s_2, \ldots, s_T >$ given an observation sequence $O =< o_1, o_2, \ldots, o_T >$ is calculated

[2] The documentation of the original BIS Tagset is available at : http://www.tdil-dc.in/tdildcMain/articles/134692Draft%20POS%20Tag%20standard.pdf

as

$$P_\Lambda(s|o) = \frac{1}{Z_o} exp(\sum_{t=1}^{T}\sum_{k} \lambda_k f_k(s_{t-1}, s_t, o, t))$$

where $f_k(s_{t-1}, s_t, o, t)$ is a transition feature function of the entire observation sequence, whose weight λ_k is to be learned via training. The values of the feature functions may range between $-\infty \ldots +\infty$, but typically they are binary.

To make all conditional probabilities sum up to 1, we must calculate a normalization factor

$$Z_o = \sum_{s} exp(\sum_{t=1}^{T}\sum_{k} \lambda_k f_k(s_{t-1}, s_t, o, t))$$

which, as in HMMs, can be obtained efficiently by dynamic programming.

The objective function to be maximized for training CRF is the penalized log-likelihood of state sequences given observation sequences :

$$L_\Lambda = \sum_{i=1}^{N} log(P_\Lambda(s^{(i)}|o^{(i)})) - \sum_{k} \frac{\lambda_k^2}{2\sigma^2}$$

Where, $\{ < o^{(i)}, s^{(i)} > \}$ is the labeled training data. The parameters λ here are set to maximize the penalized log-likelihood using Limited-memory BFGS (Sha and Pereira, 2003).

For the case of POS tagging, an observation sequence is tokens of a sentence and the state sequence is its corresponding sequence of labels or POS tags. We have used CRF++[3] for training and testing our tagger.

4.2. Evaluation

We evaluated our tagger in the following manner. We included the complete tagged corpus and calculated 10-fold cross-validation accuracy and best accuracy.

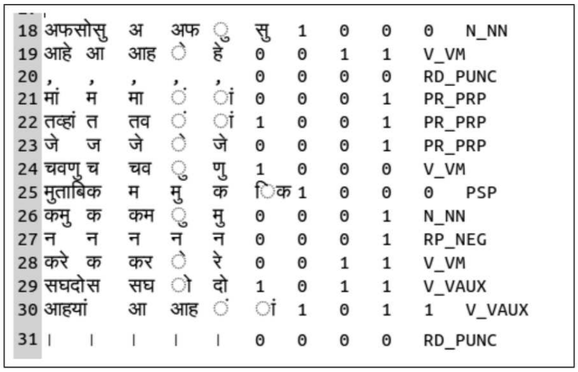
$$\text{Accuracy} = \frac{\text{Total no. of correctly tagged tokens}}{\text{Total no. of tokens}}$$

$$\text{Average Accuracy} = \frac{\text{Accuracy of all folds}}{\text{Total no. of folds}}$$

The cross-validation tests were really helpful in verifying whether our model was over-fitting and whether our test results sounded reasonable

5. Experiments, Results and Analysis

Our aim was to find the most suitable set of features for our language. We carried out our experiments with different set of features, analyzed our results and then introduced new features based on the error analysis. Some of the major experiments are described below. We would also describe the features in order, with the corresponding experiment they were introduced in.



```
अफसोसु  अ   अफ  ◌ु   सु   1   0   0   0   N_NN
आहे  आ   आह  ◌े   हे   0   0   1   1   V_VM
,    ,   ,   ,   ,   0   0   0   0   RD_PUNC
मां   म   मा   ◌ं   ◌ां  0   0   0   1   PR_PRP
तव्हां त   तव   ◌ं   ◌ां  1   0   0   1   PR_PRP
जे   ज   जे   ◌े   जे   0   0   0   1   PR_PRP
चवणु च   चव   ◌ु   णु   1   0   0   0   V_VM
मुताबिक   म   मु   क   ◌िक 1   0   0   0   PSP
कमु  क   कम   ◌ु   मु   0   0   0   1   N_NN
न    न   न   न   न   0   0   0   1   RP_NEG
करे  क   कर   ◌े   रे   0   0   1   1   V_VM
सघदोस    सघ  ◌ो   दो   1   0   1   1   V_VAUX
आहयां    आ   आह  ◌ं   ◌ां  1   0   1   1   V_VAUX
।    ।   ।   ।   ।   0   0   0   0   RD_PUNC
```

Figure 1: A tagged sentence with features: 2 suffixes, 2 prefixes and binary features.

5.1. CRF-0

We first created a simple model without using any features. We learned the best tag for a given token based on the frequency of their occurrence together. This model helped us in understanding what one could achieve with this amount of annotated data and without any linguistic knowledge to incorporate as feature in training a POS tagger. We refer to it as the simple baseline. This experiment resulted in a model with best case (out of 10-folds) accuracy of 82.35% and an average accuracy of 80%. Thus, features are important and we need them to build a better model with reasonable accuracy.

5.2. CRF-1, The Baseline

Context is an important part of POS tagging task. We used context and combination as our only feature set to train this model. Here, context refers to preceding and following tokens with respect to a token. A context window of 5 is represented as “[-2,2]” and it implies 5 uni-grams : tokens whose position is in the range of of -2 to 2 relative to the current word and the current word itself. Combination on the other hand combines the next and the current token into one token (represented as : 0/1). The bi-gram feature (represented as : B) is also a combination feature, it creates a set of unique features from the all the features of the current and the previous tokens

Our baseline model's best case (out of 10-folds) accuracy was 90.12%. We have seen that context has been used for POS tagging since the beginning. The context can only get us so far but the specialty of Indo-Aryan languages is their morphological richness, which we have not considered yet.

5.3. CRF-2, Incorporating Morphology

Morphology is the study of the internal structure of words. Affixation is a process which defines morphology of a word by attaching an affix to its root form. Prefixes and Suffixes are two kinds of affixes which we made use of to capture the morphology of the Sindhi words. We included first and last few characters of a word as its features in the data. An example is given in

[3] http://taku910.github.io/crfpp/

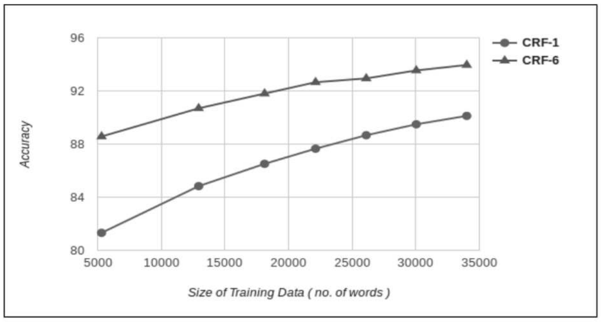


Figure 2: Plot of Accuracy v/s Size of Training Data, comparing the baseline and best model.

Figure 1, where the first two columns represent prefixes upto length 2 and next two columns represent suffixes upto length 2. We experimented with various combination of number of prefixes and suffixes and eventually got the best case accuracy of 93.73% by using 3 prefixes and 6 suffixes.

5.4. CRF-3

Words in a language are broadly classified into open class and closed class. Open class (nouns, verbs, etc.) use inflection quite extensively which makes them lengthy. We thought including word length as a feature might help in discriminating between open and closed classes.

A binary feature for word length was added to the training and testing data. Its feature value was set to 1 if the length of a word exceeded 3 characters, else 0. The feature helped in improving the tagger's best case accuracy to 93.80%. The F-1 score of Verbs (V_VM and V_VAUX) increased by 0.01 points each.

We noticed that 8.5% of Numerals (tagged as QT_QTC) were being miss-classified as Nouns. In our data we have digits in either of these closed sets : Roman [0-9] or Devanagari [०-९] script. We can use this information to help classify the numerals better.

Apart from numerals, the auxiliary verbs (V_VAUX) also belong to a closed set and yet they are one of the most hard-to-disambiguate pair. This may be due to the fact that some of these auxiliaries are also main verbs (V_VM) and an auxiliary always has a main verb in its context.

5.5. CRF-4 and CRF-5

We introduced binary features for numerals (in CRF-4) and for auxiliary verbs (in CRF-5). So, if the word belongs to the fixed set of digits or auxiliary verbs, the feature value is set to 1, else 0.

The accuracy increased to 93.85% and 93.95% using the respective features. F-1 score of QT_QTC increased to .92 and Precision of V_VAUX increased by .03 points and instances of V_VM - V_VAUX ambiguity decreased to 35.

हू	सुठो	आहे
hu	sutho	Ahe
He/That	good	is
राम	सुठो	आहे
rAm	sutho	Ahe
Ram	good	is

Table 1: Two sentences with similar context. One having a function word (hu) another has a noun (rAm).

5.6. CRF-6

We had catered to open class words earlier by incorporating their inflectional property. Similarly, we should also cater to closed class words (or function words) . Function words occur with open class words in their context. There are also instances where a similar context may not necessarily mean the presence of a closed class word. An example is depicted in Table 1. This can be a source of ambiguity. Another property of function words is that they have a very high frequency in the corpus.

We created a list of top 150 high frequency words from the corpus. We used this list to include another binary feature, whose value is set to 1 if a word belongs to this list, else 0. The result of this experiment was an increment in accuracy to 94.01%.

The results of all the above experiments are consolidated and reported in Table 2.Hence, the configuration of CRF-6 produces the best model for POS tagging Sindhi-Devanagari.

5.7. Further Observation and Analysis

POS	Precision		Recall	
	CRF-1	CRF-6	CRF-1	CRF-6
N_NN	0.82	0.89	0.91	0.94
N_NNP	0.83	0.84	0.56	0.77
V_VAUX	0.90	0.93	0.85	0.94
V_VM	0.84	0.94	0.91	0.94

Table 3: Precision and Recall based comparison between baseline and final model for morphologically rich categories.

The experiments above have shown the effect of various features on the model accuracy. An interesting fact that we noted was the effect of training data size on model accuracy. So, we compared our baseline and best model on the basis of size of training data and also observed the curve of accuracy versus data size (see Figure 2). We observed that the curve has not reached a plateau, eventually. This suggests that there's scope of further improvement in accuracy by using more training data.

6. Conclusion and Future Work

In this paper we have reported the accuracy obtained by training a CRF based model on POS annotated data of a less resourced language Sindhi. We

Model	Context	Affixes	Comb.	WL	NUM	AUX	FW	Accuracy	Avg. Accuracy
CRF-0	-	-	-	-	-	-	-	82.35	80.00
CRF-1	[-1,1]	-	0/1	-	-	-	-	90.12	87.73
CRF-2	[-1,1]	3,6	0/1 ; B	-	-	-	-	93.73	91.61
CRF-3	[-1,1]	3,6	0/1 ; B	yes	-	-	-	93.80	91.7
CRF-4	[-1,1]	3,6	0/1 ; B	yes	yes	-	-	93.85	91.73
CRF-5	[-1,1]	3,6	0/1 ; B	yes	yes	yes	-	93.95	91.75
CRF-6	[-1,1]	3,6	0/1 ; B	yes	yes	yes	yes	94.01	91.78

Table 2: Accuracy of each model and the features it was trained on. Context = range of adjacent tokens considered. Affixes = (prefixes, suffixes). Comb. = Combination features. WL = Word length. AUX = Auxiliary verbs. FW = Function words.

found that using linguistically oriented features (affixes, word length, stop words, auxiliary verbs) makes a significant impact. These features coupled with capturing context help in developing a good POS-Tagger. Then, the size of training data is the next most important factor in further enhancing the tagger.

Future work includes developing more annotated corpus by bootstrapping on this corpus using the tagger. Then, we would like to observe our accuracy on larger data and estimate the size of annotated data required for making a reasonably accurate POS tagger for a resource poor language. Also, other important tools such as Named Entity Recognizer (NER) and shallow parsers, can also be built upon this data and tagger. We would also like to compare CRFs with other models such as SVMs, TnT, TreeTagger and attempt combining them to form an ensemble tagger.

7. Acknowledgements

We would like to acknowledge a lot of people for their help and support. Pinkey Nainwani for helping us in creating and annotating the data. Chunnilal Wadhwani, for teaching us reading and writing in Perso-Arabic script. When we weren't able to find raw data, Mr. Bhagwan Babani and Mr. Gyanchand Lalwani helped us in obtaining articles, books, etc. We are also indebted to the Sindhi community for their help, support and suggestions.

8. References

Black, Ezra, Fred Jelinek, John Lafferty, Robert Mercer, and Salim Roukos, 1992. Decision tree models applied to the labeling of text with parts-of-speech. *In Proc. of Workshop on Speech and Natural Language (HLT/ACL).*

Brants, Thorsten, 2000. Tnt: A statistical part-of-speech tagger. *Proceedings of the Sixth Conference on Applied Natural Language Processing*:224–231.

Cutting, Doug, Julian Kupiec, Jan Pedersen, and Penelope Sibun, 1992. A practical part-of-speech tagger. *In Proc. of 3rd Conference on Applied Natural Language Processing.*

Daswani, C.J., 1979. Movement for the recognition of sindhi and choice of a script for sindhi. *Language Movements in India.*

Ekbal, Asif, Rejwanul Haque, and Sivaji Bandyopadhyay, 2007. Bengali part of speech tagging using conditional random field. *In Proc.of the 7th International Symposium on Natural Language Processing (SNLP-07).*

Garside, R. and N. Smith, 1997. A hybrid grammatical tagger: Claws4. *Corpus annotation: Linguistic information from computer text corpora.*

Lafferty, John, Andrew McCallum, and Fernando Pereira, 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *In Proceedings of ICML.*

Mahar, Javed Ahmed and Ghulam Qadir Memon, 2010. Rule based part of speech tagging of sindhi language. *International Conference on Signal Acquisition and Processing.*

Nongmeikapam, Kishorjit and Sivaji Bandyopadhyay, 2012. A transliteration of crf based manipuri pos tagging. *In Proc. of 2nd International Conference on Communication, Computing & Security.*

Patel, Chirag and Karthik Gali, 2008. Part-of-speech tagging for gujarati using conditional random fields. *Proceedings of the IJCNLP.*

Rahman, Mutee U and Mohammad Iqbal Bhatti, 2010. Finite state morphology and sindhi noun inflections. *In Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation, PACLIC 24, Tohoku University, Japan.*

Ratnaparkhi, A., 1996. A maximum entropy model for part-of-speech tagging. *In Proc. of EMNLP.*

Sha, Fei and Fernando Pereira, 2003. Shallow parsing with conditional random fields. *In Proc. of the NAACL-HLT, Canada.*

Shambhavi, B R and Ramakanth Kumar P, 2012. Kannada part-of-speech tagging with probablistic classifiers. *International Journal of Computer Applications.*

Shrivastava, Manish, R. Melz, Smriti Singh, K. Gupta, and Pushpak Bhattacharya, 2006. Conditional random field based pos tagger for hindi. *In Proc. of the MSPIL, Bombay.*