NE Tagging for Urdu based on Bootstrap POS Learning

Smruthi Mukund
Dept. of Computer Science and Engineering
University at Buffalo, SUNY
Amherst, NY, USA
smukund@buffalo.edu

Rohini K. Srihari
Dept. of Computer Science and Engineering
University at Buffalo, SUNY
Amherst, NY, USA
rohini@cedar.buffalo.edu

Abstract

Part of Speech (POS) tagging and Named Entity (NE) tagging have become important components of effective text analysis. In this paper, we propose a bootstrapped model that involves four levels of text processing for Urdu. We show that increasing the training data for POS learning by applying bootstrapping techniques improves NE tagging results. Our model overcomes the limitation imposed by the availability of limited ground truth data required for training a learning model. Both our POS tagging and NE tagging models are based on the Conditional Random Field (CRF) learning approach. To further enhance the performance, grammar rules and lexicon lookups are applied on the final output to correct any spurious tag assignments. We also propose a model for word boundary segmentation where a bigram HMM model is trained for character transitions among all positions in each word. The generated words are further processed using a probabilistic language model. All models use a hybrid approach that combines statistical models with hand crafted grammar rules.

1 Introduction

The work here is motivated by a desire to understand human sentiment and social behavior through analysis of verbal communication. Newspapers reflect the collective sentiments and emotions of the people and in turn the society to which they cater to. Not only do they portray an event that has taken place as is, but they also reveal details about the intensity of fear, imagination, happiness and other emotions that people express in relation to that event. Newspaper write ups, when analyzed over these factors - emotions, reactions and behavior - can give a broader perspective on the culture, beliefs and the extent to which the people in the region are tolerant towards other religions. Our final goal is to automate this kind of behavioral analysis on newspaper articles for the Urdu language. Annotated corpus that tag six basic human emotions, "happy", "fear", "sad", "surprise", "anger" and "disgust", based on the code book developed using the MPQA standards as guideline, is currently being developed. Articles from two leading Urdu newswires, BBC Urdu[1] and Jung Daily[2] form our corpus.

In order to achieve our goal, it was required to generate the basic tools needed for efficient text analysis. This includes NE tagging and its precursor, POS tagging. However, Urdu, despite being spoken by over 100 million people, (Gordon, 2005) is still a less privileged language when it comes to the availability of resources on the internet. Developing tools for a language with limited resources is a challenge, but necessary, as the volume of Urdu text on the internet is rising. Huda (2001) shows that Urdu has now gained importance on the web, making it the right time to tackle these issues.

It is useful to first examine some basic properties of Urdu and how they affect the cascade of NLP steps in text analysis. Urdu has the *nastaleeq* and *nasq* style of writing that is similar to Arabic

[1] http://www.bbc.co.uk/urdu/
[2] http://www.jang.net/urdu/

Proceedings of CLIAWS3, Third International Cross Lingual Information Access Workshop, pages 61–69,
Boulder, Colorado, June 2009.

and flows from right to left (Ahmad et al., 2001). It also adopts some of its vocabulary from Arabic. However, the grammar and semantics of the language is similar to Hindi and this makes it very different from Arabic. For effective text analysis, a thorough syntactic and semantic understanding of the language is required. Detailed grammatical analysis provided by Platts (1909) and Schmidt (1999) can be used for this purpose. The first step in the information retrieval pipeline is tokenization. Unlike English, where the word delimiter is mostly a space, Urdu is more complex. There are space insertion as well as space deletion problems. This makes tokenization a difficult task. The word segmentation model that we propose here combines the statistical approach that considers bigram transition of characters based on their positions in a word and morphological rules with lexicon lookups.

POS tagging comes next in the NLP text analysis pipeline. The accuracy of the tagging model varies, depending on the tagsets used and the domain of the ground truth data. There are two main tagsets designed for Urdu, the CRULP tagset[3] and the U1-tagset (Hardie 2003). The U1-tagset, released as a part of EMILLE[4] corpus, is based on the EAGLES standards (Leech and Wilson 1999). We decided to use the standards proposed by CRULP for the following reasons.

1. The tagset, though not as detailed as the one proposed in U1-tagset, covers all the basic requirements needed to achieve our final goal.
2. The tagged corpus provided by CRULP is newswire material, similar to our final corpus.

A person, when asked to identify an NE tagged word in a sentence would typically try to first find the word associated with a proper noun or a noun, and then assign a suitable NE tag based on the context. A similar approach is used in our model, where the learning happens on the data that is POS tagged as well as NE tagged. Features are learnt from the POS tags as well as the NE tags. The final output of our complete model returns the POS tags and NE tags associated with each word. Since we have limited data for training both the POS as well as the NE models, we propose a technique called bootstrapping that helps in maximizing the learning for efficient tagging.

The remainder of the paper is organized as follows. Section 2 discusses the resources assimilated for the work followed by tokenization and word segmentation in Section 3. Section 4 gives a detailed explanation of our model starting with a brief introduction of the learning approach used. Rules used for POS tagging and NE tagging are mentioned in subsections of Section 4. Section 5 presents the results and Section 6 concludes the paper. In each section, wherever relevant, previous work and drawbacks are presented.

2 Resources

Based on the style of writing for Urdu, different encoding standards have been proposed. *Urdu Zabta Takthi* - the national standard code page for Urdu and *Unicode* - international standard for multilingual characters are the two proposed and widely used encoding standards. BBC Urdu and Jung Daily are both encoded with Unicode standards and are good sources of data. The availability of online resources for Urdu is not as extensive as other Asian languages like Chinese and Hindi. However, Hussain (2008) has done a good job in assimilating most of the resources available on the internet. The lexicon provided as a part of the EMILLE (2003) data set for Urdu has about 200,000 words. CRL[5] has released a lexicon of 8000 words as a part of their Urdu data collection. They also provide an NE tagged data set mostly used for morphological analysis. The lexicon includes POS information as well. CRULP[6] has also provided a lexicon of 149,466 words that contains places, organizations and names of people. As part of the Urdu morphological analyzer provided by Humayoun (2007), a lexicon of about 4,500 unique words is made available. There are a few Urdu-English dictionaries available online and the first online dictionary, compiled by Siddiqi (2008), provides about 24,000 words with their meanings in English.

Getting all the resources into one single compilation is a challenge. These resources were brought

[3] http://www.crulp.org/Downloads/ling_resources/parallelcorpus/Urdu POS Tagset.pdf

[4] http://www.emille.lancs.ac.uk/

[5] http://crl.nmsu.edu/Resources/lang_res/urdu.html

[6] http://www.crulp.org/software/ling_resources/wordlist.htm

together and suitably compiled into a format that can be easily processed by Semantex (Srihari, 2008), a text extraction platform provided by Janya Inc[7]. Lists of places, organizations and names of famous personalities in Pakistan were also compiled using the Urdu-Wikipedia[8] and NationalMaster[9]. A list of most common names in Pakistan was composed by retrieving data from the various name databases available on the internet.

The word segmentation model uses the Urdu corpus released by CRULP as the training data. This dataset is well segmented. POS tagging model uses data provided by CRULP and NE tagging model uses data provided by CRL.

3 Word Segmentation and Tokenization

Urdu is a language that has both the space insertion and space deletion problems. The Urdu word segmentation problem as mentioned by Durrani (2007) is triggered by its orthographic rules and confusions about the definition of a word. Durrani summarizes effectively, all the problems associated with Urdu word segmentation. Of all the different techniques explored to achieve this objective, traditional techniques like longest and maximum matching depend mostly on the availability of a lexicon that holds all the morphological forms of a word. Such a lexicon is difficult to obtain. It is shown by Theeramunkong et al., (2001), that for a Thai segmentation system, the efficiency drops considerably (from 97% to 82%) making this approach highly lexicon dependent.

Statistical based techniques have applied probabilistic models to solve the problem of word segmentation. Bigram and trigram models are most commonly employed. Using feature based techniques for POS tagging is also very common. These techniques overcome the limitations of statistical models by considering the context around the word for specific words and collocations. There are other models that generate segments by considering word level collation as well as syllable level collocation.

However, for a language like Urdu, a model that is purely statistical will fail to yield good segmentation results. A mixed model that considers the morphological as well as semantic features of the language facilitates better performance as shown by Durrani (2007) where the word segmentation model uses a lexicon for proper nouns and a statistical model that trains over the *n*-gram probability of morphemes. Maximum matching technique is used to generate word boundaries of the orthographic words that are formed and these are later verified using the POS information. The segments thus generated are ranked and the best ones are accepted. Statistical models that consider character based, syllable based and word based probabilities have shown to perform reasonably well. The Thai segmentation problem was solved by Pornprasertkul (1994) using the character based approach. In our model, we use a combination of character based statistical approach and grammar rules with lexicon lookups to generate word boundaries.

Urdu segmentation problem can be looked at as an issue of inserting spaces between characters. All letters in Urdu, with a few exceptions, have three forms - initial, medial and final. (We do not consider the detached form for word formation). Words are written by joining the letters together and based on the position of the letter in the word, suitable forms are applied. This property of word formation is the crux of our model. The bigram probability of occurrences of each of these characters, based on their positions, is obtained by training over a properly segmented training set. For unknown characters, unknown character models for all the three position of occurrences are also trained. The probability of word occurrence is noted. Along with this, a lexicon rich enough to hold all possible common words is maintained. However, this lexicon does not contain proper nouns. A new incoming sentence that is not segmented correctly is taken and suitable word boundaries are generated by using a combination of morphological rules, lexicon lookups, bigram word probabilities and bigram HMM character model. The following probabilities are estimated and maximized at character level using the Viterbi algorithm. The following are the calculated probabilities:

(i) $P(ch_{k(medial)} \mid ch_{k-1(initial)})$ - is the probability of character *k* being in medial form given character *k*-1 is in initial form.

[7] http://www.janyainc.com/
[8] http://ur.wikipedia.com/wiki/
[9] http://www.nationmaster.com/index.php

(ii) $P(ch_{k(final)} \mid ch_{k-1(initial)})$ - is the probability of character *k* being in final form given character *k*-1 is in initial form.

(iii) $P(ch_{k(final)} \mid ch_{k-1(medial)})$ - is the probability of character *k* being in final form given character *k*-1 is in medial form.

(iv) $P(ch_{k(medial)} \mid ch_{k-1(medial)})$ - is the probability of character *k* being in medial form given character *k*-1 is in medial form.

(v) $P(ch_{k(initial)} \mid ch_{k-1(final)})$ - is the probability of character *k* being in initial form given character *k*-1 is in final form.

Each word thus formed successfully is then verified for morphological correctness. If the word is not valid morphologically, then the window is moved back over 3 characters and at every step the validity of occurrence of the word is noted. Similarly, the window is moved 3 characters ahead and the validity of the word is verified. All words formed successfully are taken and further processed using a language model that considers the bigram occurrence for each word. The unknown word probability is considered here as well. The word with maximum probability is taken as valid in the given context.

Let $< w_1 w_2 w_3 >$ be the word formed by the moving window. Then, the word selected, w_s, is given by

(vi) $$w_s = \max \begin{Bmatrix} P(w_1) \mid P(w_{prev}) \\ P(w_2) \mid P(w_{prev}) \\ P(w_3) \mid P(w_{prev}) \end{Bmatrix}$$

where w_{prev} is the previous word.

It is also noted that the number of times a transition happens from a syllable set with consonants to a syllable set with vowels, in a word, is no longer than four in most cases as noted below. This factor is also considered for terminating the Viterbi algorithm for each word.

Ir | *aad* | *ah* - three transitions

Some of the morphological rules considered while deciding the word boundaries are given below. Word boundary is formed when

1. The word ends with "ں" - *Nun Gunna*
2. The character transitions over to digits
3. Punctuations marks are encountered ('-' is also included)
4. No two '*ye*' - *choti ye* come back to back
5. No characters occur in detached form unless they are initials or abbreviations followed by a period
6. If current character is *'alif'* and the previous character is *'ee*' - *bari ye* then the word boundary occurs after '*alif*'

Some of the drawbacks seen in this model are mainly on account of improper identification of proper nouns. If a proper noun is not well segmented, the error propagates through the sentence and typically the next two or three words fail to get segmented correctly. Also, in Urdu, some words can be written in more than one ways. This mostly depends on the diacritics and ambiguity between *bari* and *choti* '*ye*'. The training data as well as the test data were not normalized before training. The model shows a precision of 83%. We realized that the efficiency of this model can be improved if phoneme level transitions were taken into consideration. Training has to be increased over more proper nouns and a lexicon for proper nouns lookup has to be maintained. Diacritics that are typically used for beautification should be removed. Words across the documents need to be normalized to one accepted format to assure uniqueness. This involves considerable amount of work and hence, in order to prevent the propagation of error into the NLP text analysis pipeline, we decided to test our subsequent models using pre-segmented data, independent of our word segmentation model.

4 Learning Approaches

A Conditional Random Field (CRF), is an undirected graphical model used for sequential learning. The tasks of POS tagging and NE tagging are both sequential learning tasks and hence this learning approach is a reasonable choice. What follows is a brief outline about CRF. Interested readers are referred to Lafferty et al., (2001), for more information on CRF.

4.1 Conditional Random Fields (CRF)

A linear chain CRF defines a single log-linear probabilistic distribution over the possible tag sequences y for a sentence x

$$p(y \mid x) = \frac{1}{Z(x)} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(t, y_t, y_{t-1}, x_t)$$

where $f_k(t, y_t, y_{t-1}, x_t)$ is typically a binary function indicating the presence of feature k, λ_k is the weight of the feature, and $Z(x)$ is a normalization function.

$$Z(x) = \sum_{y} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(t, y_t, y_{t-1}, x_t)$$

This modeling allows us to define features on states (the POS/NE tags) and edges (pairs of adjacent POS/NE tags) combined with observations (eg. words and POS tags for NE estimation). The weights of the features are determined such that they maximize the conditional log-likelihood of the training data:

$$L(\theta) = \sum_{i=1}^{N} \log\left(p_\theta(y^{(i)} \mid x^{(i)})\right).$$

For the actual implementation, CRF++[10], an open source tool that uses the CRF learning algorithm is used. The L-BFGS algorithm[11] is used for optimization.

4.2 NE Tagging using POS information

POS tagging is a precursor for all text analysis tasks. Assigning POS tags to words without any ambiguity depends on contextual information and extracting this information is a challenge. For a language like English, several techniques have been proposed that can be broadly classified into statistical, rule based and hybrid approaches (Ekbal, 2007). The general consensus is that approaches like MEMM and HMM, that work well for Hindi, would work well for Urdu as well, since Urdu is grammatically similar to Hindi (Platts, 1909). However, the linguistic and morphological rules used in the post processing steps differ from Hindi because of Urdu's borrowed vocabulary and style of writing from Arabic. Also, the requirement for such models to work well is the availability of large training data.

Building NE recognizers for languages like Urdu is difficult as there are no concepts like capitalization of characters. Also, most names of people have specific meanings associated with them and can easily be found in a dictionary with different associated meanings. Various learning approaches have been proposed for this task, HMM based learning approach (Bikel et al., 1999), Maximum Entropy Approach (Borthwick, 1999) and CRF approach (McCallum, 2003) are the most popular. Ashish et al., (2009) show an SVM based approach also works well for such tasks. To overcome the problem of limited data availability, we present a method to increase the amount of training data that is available, by using a technique called bootstrapping.

We do not have a training corpus that is manually tagged for both POS and NE. Our training data consists of two different datasets. The dataset used for POS tagging is provided by CRULP and is tagged using their tagset. The dataset used for NE tagging is provided by CRL as a part of their Urdu resource package. The CRL tagset consists of `LOCATION`, `PERSON`, `ORGANIZATION`, `DATE` and `TIME` tags. We use only the first three tags in this work.

Our aim is to achieve effective POS tagging and NE tagging by maximizing the use of the available training data. The CRULP dataset (which we call $dataset_{\text{POS}}$) is a corpus of 150,000 words that are only POS tagged and the CRL dataset (which we call $dataset_{\text{NE}}$) is a corpus of 50,000 words that are only NE tagged. First, we trained a CRF model on $dataset_{\text{NE}}$ that uses only the NE information to perform NE recognition. This one stage model was not effective due to the sparseness of the NE tags in the dataset. The model requires more data while training. The obvious and frequently tried approach (Thamar, 2004) is to use the POS information.

Figure 1 shows a two stage model that uses POS information to perform NE tagging. The first stage POS_A performs POS tagging by using a CRF trained model to assign POS tags to each word in a sentence of $dataset_{\text{NE}}$. The second stage NE_A performs NE tagging by using another CRF trained model that uses both the POS information as well

[10] http://crfpp.sourceforge.net/
[11] http://www.mcs.anl.gov/index.php

as the NE information, to perform effective NE tagging.

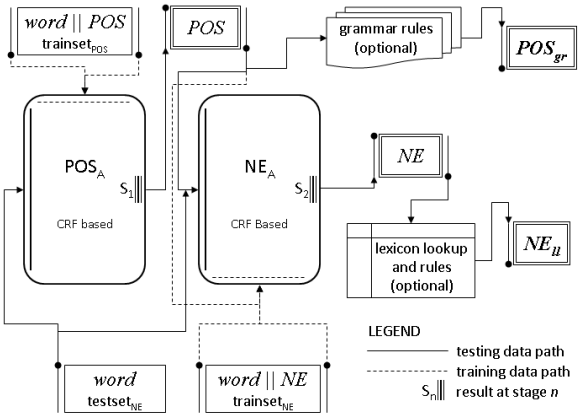

Figure 1. Two stage model for NE tagging using POS information

However, although the accuracy of NE tagging improved over the one stage model, there was scope for further improvement. It is obvious that all the NE tagged words should have the proper noun (NNP) POS tag associated. But, when POS tags were generated for the NE tagged ground truth data in $dataset_{NE}$, most of the words were either tagged as adjectives (JJ) or common nouns (NN). Most tags that come after case markers (CM) were adjectives (JJ) in the training data. Very few accounted for proper nouns after case markers. This adversely affected the NE tagger output. It was also noticed that the POS tagger tagged most of the proper nouns (NNP) as common nouns (NN) because of the sparseness of the proper noun tag in the POS ground truth data set $dataset_{POS}$. This observation made us look to bootstrapping techniques for effective learning.

We propose a four stage model as shown in Figure 2, for NE tagging. Three of the stages are trained using the CRF learning approach and one stage uses a rule based approach. All four stages are trained using unigram features on tags and words and bigram features on tags. The POS tagged dataset, $dataset_{POS}$, consists of words and associated POS tags and the NE tagged dataset, $dataset_{NE}$, consists of words and associated NE tags. We divide both datasets into training and testing partitions. $dataset_{POS}$ is divided into $trainset_{POS}$ and $testset_{POS}$ and $dataset_{NE}$ is divided into $trainset_{NE}$ and $testset_{NE}$.

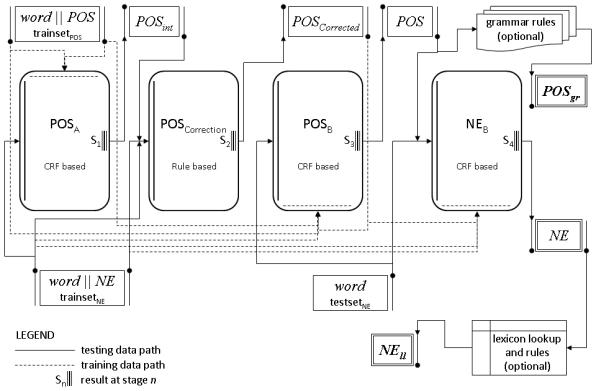

Figure 2. Four stage model for NE tagging using POS information with bootstrapping

In the model shown in Figure 2, POS_A stage is a CRF based stage that is trained using $trainset_{POS}$. Once trained, the POS_A stage takes as input a sentence and generates the associated POS tag for each word in that sentence.

In order to increase the NNP tag associations to improve NE tagging, we generate POS tags for the NE training data in $trainset_{NE}$ using the POS_A stage. The POS tags generated at the POS_A stage are called POS_{int}. The $POS_{correction}$ stage takes as input $trainset_{NE}$ along with its associated POS tags, POS_{int}. At this stage, correction rules - that change the POS tags of NE associated words to proper noun (NNP), assign Case Markers (CM) before and after the NE tags and verify proper tagging of Cardinals (CD) - are applied. The corrected POS tags are called $POS_{corrected}$. A consolidated POS training set consisting of entries from both $trainset_{POS}$ and $trainset_{NE}$ (with $POS_{corrected}$ generated as output from the $POS_{correction}$ stage) is used to train the CRF based POS_B stage. This stage is the final POS tagging stage. Test data consisting of sentences (words) from $testset_{NE}$ is sent as input to stage POS_B and the output generated at stage POS_B is the POS tag associated with each input word of a sentence. The NE_B stage is a CRF based NE tagger that is trained on a dataset consisting of word and associated NE tags from $trainset_{NE}$ and associated POS tags from $POS_{corrected}$. This stage learns from the POS information and the NE information provided in the training data. Once trained, the NE_B stage takes as input words from $testset_{NE}$ and associated POS tags (obtained at stage POS_B) and generates NE tags.

The domain we are interested in is newswire material, and these articles are written in the "jour-

nalistic" or "news writing" style[12]. The articles are objective and follow a Subject-Object-Verb structure. Related information is usually presented within close sentence proximity. This makes it possible to hand-craft grammar rules for the discovery of NE tags with fine granularity. The final POS tagged and NE tagged data generated as outputs at stage POS_B and stage NE_B respectively of the four stage model, are processed using rules and lexicon lookups to further improve the overall tagging accuracy of the model. Rules used are mostly domain specific. The rules were applied to the model using Semantex.

4.3 Rules for POS Tagging

1. Our model tags all the Question Words (QW) like 'کیا' - *kya* as pronoun (PR). All such occurrences are assigned QW tag.
2. If the word is 'کیا' – *kya* and the previous tag is an adjective (JJ) and the next tag is a phrase marker (PM) then assign a light verb tag (VBL) else assign a verb (VB) tag to the word.
3. It was observed that there were spurious instances of proper nouns getting tagged as nouns. In order to correct this error, if a word ends with any of the characters shown below, and the word was tagged as a noun, then the tag on the word was changed to a proper noun.
   'ی', 'ا', 'ے', 'پن', 'یا', 'ہٹ', 'گاہ', 'وٹ', 'گی'
4. All valid cardinals were tagged as nouns or proper nouns by the model. This was resolved by looking for a digit in the string.

4.4 Rules for NE Tagging

1. Words like "کورٹ" (court), "بیورو" (bureau), "فوج" (army) etc. are looked up. If there are any nouns or proper nouns above these within a window of two, then the tag on this word is ORGANIZATION.
2. Words like "تنظیم" (organization), "آرمی" are marked ORGANIZATION if the previous word is a proper noun.
3. Lexicon look up for names of places is performed and the POS tag of the next word that is found is checked. If this tag is a Case Marker (CM) with a feminine gender, like "کے" (main) or "میں", then the word is marked with a LOCATION tag.
4. If a proper noun that is selected ends with a suffix "pur", "bad, "dad" and has the same constraint as mentioned in rule 3, then the LOCATION tag is assigned to it as well.

5 Results

The NE tagging performance, for both the two stage model and the four stage model, are evaluated using Precision (P), Recall (R) and F-Score (FS) metrics, the equations for which are given below.

(vii) $$P = \frac{\text{No. of correctly tagged NEs}}{\text{No. of tagged NEs}}$$

(viii) $$R = \frac{\text{No. of tagged NEs}}{\text{Total no. of NEs in test set}}$$

(ix) $$FS = \frac{2RP}{R+P}$$

We performed a 10 fold cross validation test to determine the performance of the model. The dataset is divided into 10 subsets of approximately equal size. One subset is withheld for testing and the remaining 9 subsets are used for training. This process is repeated for all 10 subsets and an average result is computed. The 10 fold validation test for NE tagging was performed for both the two stage as well as the four stage models.

	Two Stage Model			Four Stage Model		
Set	**P**	**R**	**FS**	**P**	**R**	**FS**
1	48.09	73.25	58.06	60.54	78.7	68.44
2	38.94	72.42	50.65	60.29	80.46	68.93
3	56.98	74.38	64.53	60.54	79.74	68.83
4	38.44	78.05	51.51	60.54	80.79	69.21
5	32.29	75.91	45.31	60.79	80.34	69.21
6	44.82	88.02	59.4	59.31	79.93	68.09
7	45.75	69.75	55.26	61.04	81.73	69.89
8	43.52	71.5	54.11	60.05	80.36	68.74
9	44.64	81.97	57.8	59.93	81.09	68.92
10	44.17	78.18	56.45	60.67	79.22	68.72
Avg	**43.764**	**76.343**	**55.308**	**60.37**	**80.236**	**68.898**

Table 1. NE tagging results for the two stage and four stage models

It can be seen from Table 1 that the four stage model outperforms the two stage model with the

[12] http://en.wikipedia.org/wiki/News_writing

average F-Score being 55.31% for the two stage model and 68.89% for the four stage model.

Table 2 shows the POS tagging results for stages POS_A and POS_B. The POS_B stage performs marginally better than the POS_A stage.

POS_A Results		POS_B Results	
Set	**P**	**Set**	**P**
1	84.38	**1**	83.97
2	89.32	**2**	89.84
3	88.09	**3**	88.48
4	89.45	**4**	89.66
5	89.66	**5**	89.76
6	90.57	**6**	90.63
7	81.1	**7**	89.24
8	89.47	**8**	89.5
9	89	**9**	89.12
10	89.12	**10**	89.25
Avg	**88.016**	**Avg**	**88.945**

Table 2. POS tagging results for the two stage (POS_A) and four stage (POS_B) models

Although for POS tagging, the improvement is not very significant between the two models, tags like light verbs (VBLI), auxiliary verbs (AUXA and AUXT), adjectives (JJ), demonstratives (DM) and nouns (NN, NNC, NNCM, NNCR) get tagged with higher accuracy in the four stage model as shown in Table 3. This improvement becomes evident in the NE test set. Unfortunately, since this data has no associated POS tagged ground truth, the results cannot be quantified. The $trainset_{POS}$ training data had very few instances of proper nouns (NNP) occurring after case markers (CM) and so most of the proper nouns were getting tagged as either adjectives (JJ) or common nouns (NN). After providing more training data to stage POS_B, the model could effectively learn proper nouns. Spurious tagging of adjectives (JJ) and common nouns (NN) reduced while more proper nouns (NNP, NNPC) were tagged accurately and this allowed the NE stage to apply its learning efficiently to the NE test set thereby improving the NE tagging results.

The two stage model tagged 238 NE tagged words as proper nouns out of 403 NE words. The four stage model tagged 340 NE tagged words as proper nouns out of 403 NE words. The four stage model shows an improvement of 25.3% over the two stage model. The results reported for NE and POS tagging models are without considering rules or lexicon lookups.

POS_A Output		POS_B Output	
Tag	**FS**	**Tag**	**FS**
AUXA	0.801	AUXA	0.816
AUXT	0.872	AUXT	0.898
DM	0.48	DM	0.521
JJ	0.751	JJ	0.765
NN	0.85	NN	0.858
NNC	0.537	NNC	0.549
NNCM	0.909	NNCM	0.923
NNCR	0.496	NNCR	0.51
RB	0.785	RB	0.834
VBLI	0.67	VBLI	0.693
VBT	0.553	VBT	0.586

Table 3. POS tagging results for stages POS_A and POS_B

In order to further improve the POS tagged results and NE tagged results, the rules mentioned in sections 4.3 and 4.4 and lexicon lookups were applied. Table 4 shows the result for NE tagging with an overall F-Score of 74.67%

	NE_A Output		
Tag	**P**	**R**	**FS**
LOCATION	0.78	0.793	0.786
ORGANIZATION	0.775	0.731	0.752
PERSON	0.894	0.595	0.714

Table 4. NE tagging results after applying rules for test results in Table 1

6. Conclusion and Future Work

This work was undertaken as a precursor to achieve our final objective as discussed in Section 1. The basic idea here is to increase the size of the available training data, by using bootstrapping, so as to maximize learning for NE tagging. The proposed four stage model shows an F-Score of 68.9% for NE tagging which is much higher than that obtained by the simple two stage model.

A lot of avenues remain to be explored to further improve the performance of the model. One approach would be to use the bootstrapping technique for NE data as well. However, the rules required can be complicated. More hand crafted rules and detailed lexicon lookups can result in better NE tagging. We have also noticed certain ambiguities in tagging PERSON and LOCATION. Rules that resolve this ambiguity can be explored.

References

Raymond G. Gordon Jr. (ed.). 2005. *Ethnologue: Languages of the World, Fifteenth edition.* Dallas, TX.: SIL International

Kashif Huda. 2001. *An Overview of Urdu on the Web*. Annual of Urdu Studies Vol 20.

Zaheer Ahmad, Jehanzeb Khan Orakzai, Inam Shamsher, Awais Adnan. 2007. *Urdu Nastaleeq Character Recognition.* Proceedings of World Academy of Science, Engineering and Technology. Volume 26, ISSN 2070-3740.

John T. Platts. 1967. *A grammar of the Hindustani or Urdu language.* Munshiram Manoharlal Delhi.

R. L. Schmidt. 1999. *Urdu: an essential grammar*. London: Routledge.

Sarmad Hussain. 2008. *Resources for Urdu Language Processing*. The 6th Workshop on Asian Language Resources.

P. Baker, A. Hardie, T. McEnery, B.D. Jayaram. 2003. *Corpus Data for South Asian Language Processing*. Proceedings of the 10th Annual Workshop for South Asian Language Processing, EACL.

M. Humayoun, H. Hammarstrm, A. Ranta. 2007. *Urdu Morphology, Orthography and Lexicon Extraction.* CAASL-2: The Second Workshop on Computational Approaches to Arabic Script-based Languages, LSA 2007 Linguistic Institute, Stanford University.

Waseem Siddiqi, Shahab Alam. 2008. Online Urdu-English and English-Urdu dictionary.

N. Durrani. 2007. *Typology of Word and Automatic Word Segmentation in Urdu Text Corpus.* National University of Computer and Emerging Sciences, Lahore, Pakistan.

T. Theeramunkong, S. Usanavasin. 2001. *Non-Dictionary Based Thai Word Segmentation Using decision trees.* In proceedings of the First International Conference on Human Language Technology Research, San Diego, California, USA.

A. Pornprasertkul. 1994. *Thai Syntactic Analysis*. Ph.D Thesis, Asian Institute of Technology.

Ismat Javed. 1981.قواعد اردو نئ. Taraqqi Urdu Bureau, New Delhi.

Abdul M. Haq. 1987.نحو و صرف اردو. Amjuman-e-Taraqqi Urdu (Hindi).

Hassan Sajjad. 2007. *Statistical Part of Speech Tagger for Urdu.* National University of Computer and Emerging Sciences, Lahore, Pakistan.

John D. Lafferty, Andrew McCallum, Fernando C.N. Pereira. 2001. *Conditional Random Fields: ProbabilisticModels for Segmenting and Labeling Sequence Data*. Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282-289.

John Chen. 2006. *How to use Sequence Tagger.* Semantex Documentation, Janya Inc.

Bikel, D.M., Schwartz, R.L., Weischedel, R.M.1999. *An Algorithm that Learns What's in a Name*. Machine Learning 34(1-3), pp. 211–231.

Borthwick, A. 1999. *Maximum Entropy Approach to Named Entity Recognition*. PhD thesis, New York University.

McCallum, A., Li, W. 2003. *Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons*. In Proceedings of CoNLL.

A. Hardie. 2003. *Developing a tagset for automated part-of-speech tagging in Urdu*. Department of Linguistics and Modern English Language, University of Lancaster.

Leech, G and Wilson, A. 1999. *Standards for tagsets. Edited version of EAGLES Recommendations for the Morphosyntactic Annotation of Corpora*. In van Halteren, H (ed.) Syntactic wordclass tagging. Dordrecht: Kluwer Academic Publishers.

Awaghad Ashish Krishnarao, Himanshu Gahlot, Amit Srinet and D. S. Kushwaha. 2009. *A Comparative Study of Named Entity Recognition for Hindi Using Sequential Learning Algorithms*. In IEEE International Advance Computing Conference (IACC '09), Thapar University, India. March 6-7.

Thamar Solario. 2004. *Improvement of Named Entity Tagging by Machine Learning*, Technical Report CCC-04-004, Coordinacin de Ciencias Computationales.

Ekbal, A. and Bandyopadhyay, S. 2007. *A Hidden Markov Model Based Named Entity Recognition System: Bengali and Hindi as Case Studies*. Springer LNCS, Vol. 4815, pp. 545.

R. K. Srihari, W. Li, C. Niu and T. Cornell,"InfoXtract: A Customizable Intermediate Level Information Extraction Engine," *Journal of Natural Language Engineering,* Cambridge U. Press, 14(1), 2008, pp..33-69.