Named Entity Recognition System for Postpositional Languages: Urdu as a Case Study

Muhammad Kamran Malik
Punjab University College of Information Technology
University of the Punjab
Lahore, Pakistan

Syed Mansoor Sarwar
Punjab University College of Information Technology
University of the Punjab
Lahore, Pakistan

***Abstract*—Named Entity Recognition and Classification is the process of identifying named entities and classifying them into one of the classes like person name, organization name, location name, etc. In this paper, we propose a tagging scheme Begin Inside Last -2 (BIL2) for the Subject Object Verb (SOV) languages that contain postposition. We use the Urdu language as a case study. We compare the F-measure values obtained for the tagging schemes IO, BIO2, BILOU and BIL2 using Hidden Markov Model (HMM) and Conditional Random Field (CRF). The BIL2 tagging scheme results are better than the other three tagging schemes using the same parameters including bigram and context window. With HMM, the F-measure values for IO, BIO2, BILOU, and BIL2 are 44.87%, 44.88%, 45.14%, and 45.88%, respectively. With CRF, the F-measure values for IO, BIO2, BILOU, and BIL2 are 35.13%, 35.90%, 37.85%, and 38.39%, respectively. The F-measure values for BIL2 are better than those of previously reported techniques**

Keywords—IOB tagging; BIO tagging; BILOU tagging; IOE tagging; BIL2 tagging; NER for Resource-poor languages

I. Introduction

Named Entity Recognition and Classification (NERC) is a process of identifying categories and classifying them into different groups, for example, person names, location names, organization names, quantities, and date. An NERC system is used in many domains including Information Extraction (IE), Machine Translation (MT), and many other Natural Language Processing (NLP) applications.

There are several ways to automatically identify Named Entities (NEs) in unstructured data. We briefly discuss these approaches.

A. Rule Based approaches

In such approaches, language experts write rules by studying the given text. These rules are used to extract NEs from the text. The drawback of these approaches is that they require in-depth linguistic knowledge.

B. Supervised Learning approaches

A large amount of tagged data is a prerequisite for using this approach. People usually tag data manually and then use this data to train a model. Different supervised machine learning algorithms including Hidden Markov Model (HMM) [2], Decision Trees (DT), Maximum Entropy (ME), Support Vector Machine (SVM), and Conditional Random Fields (CRF) [8] are used to learn patterns or rules from the tagged data.

C. Semi-Supervised Learning approaches

In these approaches, a small degree of supervision is required as compared to the supervised learning approaches that require full supervision. The main technique used in the semi-supervised learning algorithms is called "bootstrapping". For bootstrapping, a set of manually annotated data, known as seeds, is used for starting the learning process and the system learns rules from this data. These rules are then used to annotate more data. Wrong annotations are corrected manually and corrected data is again used in learning additional rules.

D. Unsupervised Learning approaches

In these approaches, NEs are grouped on the basis of contextual similarity. Different lexical resources such as Wordnet may be used for achieving better results. In such approaches, no supervision is required. Clustering is the primary technique used in the unsupervised learning algorithms to identify NEs of the same types.

In supervised learning, tagged data is required for training and testing. Multiple tagging schemes including IO, BIO, BIO2, IOE, IOE2, and BILOU exist to tag data for NEs. We suggest the use of Begin Inside Last 2 (BIL2) tagging scheme for postpositional languages including Urdu, Japanese, and Hindi. Our hypothesis is that in postpositional languages postposition plays a vital role in the decision making process for identifying NEs. For example, in "Ali (NE) nay (postposition)" and "Muhammad Ali (NE) nay (postposition)," the word "nay" is key to deciding if the preceding word is an NE or not. Only BIL2 tagging scheme tries to capture this behavior, as shown in Table 2. Based on our literature review, we have not seen the use of this technique for postpositional languages. In this paper, we compare the performance of BIL2 with IO, BIO2 and BILOU.

The structure of the paper is as follows. Section 2 describes the related work. Section 3 describes the tagging problem and two machine learning algorithms used to handle the tagging problem. Sections 4 and 5 describe the Urdu language issues and data collection process used for experimentation. Section 6 describes the details and results of experimentation. In Sections 7 and 8, we make conclusion and briefly describe future work.

II. Related Work

For NER chunking and Semantic Role Labeling (SRL) usually two types of tagging schemes are used: Inside/Outside and Start/End. [9] introduces the Inside/Outside representation to solve the Noun Phrase (NP) chunking problem. Three tags

are used to identify chunks: 'I', 'O', and 'B'. 'I' means token is inside of the chunk, 'O' means token is outside of the chunk, and B' means token is the beginning of a chunk, immediately following the previous chunk. [11] introduces three new alternate tagging schemes, i.e., IOB2, IOE1, and IOE2, and named IOB1 as the Ramshaw tagging scheme. [13] uses the Start/End tagging scheme that has been used to solve the Japanese NER task, and uses the IOBES (also known as BILOU) tagging scheme.

[10] shows that the choice of NE tags significantly impacts the results of NER. 90.8% F-measure has been reported for an English NER system using the BILOU tagging scheme, which was best result reported at the time on the CoNLL-2003 NER shared task.

[12] also uses two frequently used tagging schemes, BIO and BILOU, for NER of the Estonian language. The results of experiments described in the paper show that BILOU outperformed BIO, and F-measure values of 86.6% and 87% were achieved on BIO and BILOU, respectively.

[4] uses three different variations of the IOB tagging scheme, IOBE, IOBES, and $IOB_{12}E$, to extract names of chemical compounds and drugs. IOBE uses four tags Begin, Inside, Outside, and End, whereas the IOBES and $IOB_{12}E$ schemes use five tags.

[7] uses the IO, IOB, IOB2, IOE, IOE2 and IOBES tagging schemes to show results on the Conference on Computational Natural Language Learning (CoNLL) dataset. They used Conditional Markov Model (CMM) to calculate F-measure. The paper shows that IOE2 and IOBES yielded better results, with F-measure values of approximately 84% and 85% for IOBES and IOE2, respectively.

[3] discusses different tagging schemes including IOB, IOB2, IOE, and IOBES for Chunking, NER, and SLR purposes.

[6] uses Support Vector Machine (SVM) for the chunking of the English language. The paper describes the use IOB, IOB2, IOE, IOE2, and IOBES tagging schemes to identify chunks. Of all these schemes, IOE2 produced best results.

III. The Tagging Problem

NER is considered a sequence-labeling problem where we want to determine a vector $z = \{z_0, z_1, \dots z_T\}$ of random variables given an observed vector $\boldsymbol{X} = \{x_0, x_1, \dots x_T\}$. Each variable z_s is the NE of the word at position s, and the input **X** is divided into feature vectors. Each x_s contains various pieces of information about the word at position s, including its identity, orthographic features such as prefixes and suffixes, membership in the domain-specific lexicons, and information in the semantic databases such as Word-Net.

Let $x_{1:n}$ be the sequence of words in a sentence in the Urdu language, and $z_{1:n}$ be the NE against each word, i.e., Person, Organization, Location, etc. Let X_s be the set of all possible sentences that can be formed from the words in set X. Let S be a sequence of words (i.e., a sentence) from $x_{1:n}$ such that $S \epsilon X_s$ with x_i be the i_{th} word in S and $z_{1:n}$ be the sequence of NEs for these words, with z_i being the i[th] NE in the sequence. Now, we define the tagging problem for finding the most probable NE sequence $z_{1:n}$ for the word sequence $x_{1:n}$. More formally,

$$argmax_{z_{1:n} \epsilon z} \; P(z_{1:n} \mid x_{1:n}) \quad (1)$$

In this expression, we want to find the NE tag sequence that gives maximum probability of NE tags sequence for an Urdu sentence.

A. Hidden Markov Model (HMM)

In HMM, we have two set of states and a triple (π, A, B). The first element in the triple is a set of observable states, that is, the input sentence or word sequence $X = \{x_{1:n}\}$ such that $X \epsilon X_s$ with x_i being the i[th] word in X. The second element is the set of hidden states that is represented by NE $z_{1:n}$ for the word sequence $x_{1:n}$ with z_i being the i[th] NE in the sequence. Each NE represents one of the hidden states in HMM. In the triple (π, A, B), we define π as the initialization vector containing the initial probabilities of all NEs z_i starting an NE sequence. We define A as a matrix of probabilities (transition or prior probabilities) when the underlying Markov Process transitions from one state (i.e., NE) to another. We define B as a matrix of probabilities (emission or likelihood probabilities) of generating the word sequence $x_{1:n}$ from the underlying NE sequence $z_{1:n}$, i.e., the probability of generating (or emitting) a word x_i once the underlying Markov Process has entered a state x_i. We learn the triple (π, A, B) from our Urdu NE training data.

HMM defines the joint probability distribution over a word sequence paired with an NE sequence as

$$P(x_{1:n}, z_{1:n}) \quad (2)$$

The output of HMM is a tag sequence that maximizes this joint probability distribution, expressed as

$$argmax_{z_{1:n} \epsilon z} \; P(x_{1:n}, z_{1:n}) \quad (3)$$

To model this joint probability we consider our basic NE problem from Equation (1) as

$$argmax_{z_{1:n} \epsilon z} \; P(z_{1:n} | x_{1:n}) \quad (4)$$

Bayes Rule of probability dictates us that we can calculate the probability of $(z_{1:n}|x_{1:n})$ if we know the probability of $(x_{1:n}|z_{1:n})$. It says

$$P(z_{1:n}|x_{1:n}) = \frac{P(z_{1:n})P(x_{1:n}|z_{1:n})}{P(x_{1:n})} \quad (5)$$

By applying Bayes Rule to Equation 3 we get

$$argmax_{z_{1:n} \epsilon z} \; \frac{P(z_{1:n})P(x_{1:n}|z_{1:n})}{P(x_{1:n})} \quad (6)$$

We drop the denominator for being the constant for all NEs and hence Equation 6 becomes

$$argmax_{z_{1:n} \epsilon z} P(z_{1:n})P(x_{1:n}|z_{1:n}) \quad (7)$$

This means that for each NE sequence we need to calculate the product of likelihood probability $P(x_{1:n}|z_{1:n})$ and prior probability $P(z_{1:n})$. We make two simplifying assumptions to estimate the probability of the NE sequence. The first

assumption says that the probability of a word is dependent only on its own underlying NE.

$$P(x_{1:n}|z_{1:n}) \approx \prod_{i=1}^{n} P(x_i|z_i) \quad (8)$$

Since we have used both Bigram and Trigram HMM to formulate our results, therefore, for Bigram HMM we assume that the probability of an NE is dependent only on the previous NE (First Order Markov Assumption). Thus, $P(z_{1:n})$ is expressed as shown below.

$$P(z_{1:n}) \approx \prod_{i=1}^{n} P(z_i|z_{i-1}) \quad (9)$$

For Trigram HMM we assume that the probability of an NE is dependent only on the previous two NEs (Second Order Markov Assumption). Thus, Equation (9) may be expressed as given below.

$$P(z_{1:n}) \approx \prod_{i=1}^{n} P(z_i|z_{i-2}, z_{i-1}) \quad (10)$$

With these two assumptions, we can rewrite Equation (2) as

$$P(x_{1:n}, z_{1:n}) \approx \prod_{i=1}^{n} P(z_i|z_{i-1}) \prod_{i=1}^{n} P(x_i|z_i) \quad (11)$$

Where $P(z|z_{i-2})$ and $P(z_i|z_{i-2}, z_{i-1})$ are called the Bigram and Trigram parameters, respectively, and $P(x_i|z_i)$ is called the emission parameter of HMM.

B. Conditional Random Field (CRF)

Let $x_{1:n}$ be a sequence of words in an Urdu language sentence with $z_{1:n}$ NEs against each word, i.e., Person, Organization, Location, and Other. A linear chain CRF defines a conditional probability as

$$P(z_{1:n}|x_{1:n}) = \frac{1}{Z} exp^{(\sum_{n=1}^{N}\sum_{i=1}^{F}\lambda_i f_i(z_{n-1}, z_n, x_{1:n}, n)} \quad (12)$$

The scalar Z is the normalization factor. Z is defined as

$$Z = \sum_{z1:n} exp^{(\sum_{n=1}^{N}\sum_{i=1}^{F}\lambda_i f_i(z_{n-1}, z_n, x_{1:n}, n)} \quad (13)$$

In the exp() function, all weighted feature functions are summed against each word and for each word values are summed to compute the total score for the sentence. The scalar λi is the weight for the features *fi*(). The $\lambda i's$ are the parameters of CRF model and must be learned.

Feature function: In CRF, the feature function is the key component that consists of the current tag, previous tag, complete input sentence, and current position in the sentence. The output of the feature function is a real value. The general form of a feature function is $f_i(z_{n-1}z_n, x_{1:n}, n)$.

For example, we can define a feature function that produces binary values: it is one (1) if the current word is Ahmad, and if the current state z_n is PERSON:

$$f_1(z_{n-1}, z_n, x_{1:n}, n) = \begin{cases} 1 \; if \; z_n = PERSON \; and \; x_n = Ahmad \\ 0 \; otherwise \end{cases} \quad (14)$$

Depending upon the corresponding weight $\lambda 1 > 0$, f_1 is active only when the word Ahmad is seen and its tag is PERSON, and it increases the probability of the tag sequence $z_{1:N}$. It means that the preferred tag for Ahmad is PERSON. If $\lambda 1 < 0$, then CRF tries to avoid the tag PERSON for Ahmed. Finally, if $\lambda 1 = 0$, it means that this feature has no effect.

Another example of feature is

$$f_2(z_{n-1}, z_n, x_{1:N}, n) = \begin{cases} 1 \; if \; z_n = PERSON \; and \; x_{n+1} = "nay" \\ 0 \; otherwise \end{cases} \quad (15)$$

The value of this feature is 1 when the current tag is PERSON and the next word in Urdu sentence is 'nay'. If this pattern is found in the training data then $\lambda 2$ will be positive. Furthermore, note that f_1 and f_2 can both be active for a sentence like "Ahmad nay khaa (Ahmad said)". This is an example of overlapping features.

IV. Issues with the Urdu Language

Following are the issue of Urdu language:

1) In Urdu, there is no concept of capitalization, which is a major clue of NEs.

2) The Urdu language is Agglutinative in nature, i.e., by adding additional features to a word more complex words can be formed.

3) Urdu is free word-order language, i.e., a sentence can be written using Subject-Object- Verb or Object-Subject-Verb.

4) Very few reliable gazetteers are available for the Urdu language.

5) In the Urdu language, words are written sometimes with diacritic and sometimes without diacritic, causing multiple variations of single word.

6) Urdu is called a "Lashkari" language, i.e., it contains words of different languages including those of Arabic, Persian, and English.

7) Researchers in the field of natural language processing have not spend much time in studying the Urdu language.

8) In the Urdu language, there is the issue of word segmentation.

9) Urdu has the problem of lack of character level standardization and spelling variation.

*10)*In the Urdu language, depending upon the context, there is a large number of words that can be considered as common nouns as well as proper nouns (i.e., candidate for NE). For example, Shan, Kamran, Fazal, Kiran, Aftab, Manzoor, etc. can be NEs, i.e., Person as well as common nouns. The context may help in identifying proper nouns against common nouns but due to no concept of capitalization,

disambiguation becomes harder than that in the English language.

*11)*There are multiple ways of representing abbreviations in Urdu.

*12)*There is a serious lack of labeled data in Urdu required for machine learning.

*13)*There is a huge variation in the number formats in Urdu, for example, ghiara (11), bara (12), taira (13), ikkees (21), baaees (22), etc.

V. Data Collection

We took the corpus for our experiments from IJCNLP-08 NERSSEAL shared tasks datasets. For annotation, the first step was to identify whether a word is an NE or not. For example, the word "Fazal" is an NE or not depend on the context. Since Urdu does not have the concept of capitalization, therefore, in the sentence "Us per khuda ka fazal hai (he has the blessing of God)", fazal (blessing) is not an NE, whereas in the sentence "fazal aik laek talabilm hai (fazal is a bright student)", fazal is an NE (PERSON). The next step was to tag maximal entities. For example, "Quaid-e-Azam Library should be tagged as Location. It should not be marked "Quaid-e-Azam" as Person.

We divided the corpus into two sets: training and testing. The following are the details of Urdu that were used in the shared task. The Urdu text was partially taken from the news corpus and partially from other sources. The counts of all NEs used in training and testing are given in the Table 1.

We used 12 tags for tagging the dataset. The details of the tagset are given below:

- NEP (Person): 'Quaid-e-Azam Muhammad Ali Jannah' or simply 'Quaid-e-Azam', or 'Allama Iqbal'
- NED (Designation): 'Prime Minister', 'President' (as in 'President Musharaf'), or 'General' (as in 'General Raheel')
- NEO (Organization): 'State Bank of Pakistan', 'DELL', 'Al Qaida', or 'The Ministry of Defense'
- NEA (Abbreviation): 'PU' (or P.U.), 'CRF', 'AJK', or 'LTV'
- NEB (Brand): 'Pepsi' or 'Windows'
- NETP (Title-Person): 'Mr.', 'Sir', or 'Field Marshall'
- NETO (Title-Object): 'The Seven Year Itch', 'American Beauty', '1984' (as in '1984 by George Orwell'), or 'One Hundred Years of Solitude'
- NEL (Location): 'Lahore', 'Islamabad', or 'Punjab'
- NETI (Time): '19 May', '1965', or '6:00 pm'
- NEN (Number): 'Fifty-five', '3.50', or 'ten lac'
- NEM (Measure): '10 kg', '32 MB', or 'five years'
- NETE (Terms): 'Horticulture', 'Conditional Random Fields', 'Sociolinguistics', or 'The Butterfly Effect'

TABLE I. Statistics about Urdu Training and Testing Data

NE	Training Data	Testing Data
NEP	365	145
NED	98	41
NEO	155	40
NEA	39	3
NEB	9	18
NETP	36	15
NETO	4	147
NEL	1118	468
NETI	279	59
NEN	310	47
NEM	140	40
NETE	30	4
NEs	2584	1027
Words	35447	12805
Sentences	1508	498

VI. Methodology

We assessed the performance of our system using precision, recall, and F1-measure. We used the BIL2 tagging scheme for Subject Object Verb (SOV) ordered languages that usually contain postposition instead of preposition. Table 2 gives details for the IO, BIO2, BILOU and BIL2 tagging for the example "Syed Mansoor Sarwar worked at Punjab University Lahore".

TABLE II. Examples of Different Tagging Scheme

Words	IO	BIO2	BILOU	BIL2
سید (Syed)	I-PER	B-PER	B-PER	B-PER
منصور (Mansoor)	I-PER	I-PER	I-PER	I-PER
سرور (Sarwar)	I-PER	I-PER	L-PER	L-PER
نے (nay)	O	O	O	O
لاهور (Lahore)	I-LOC	B-LOC	U-LOC	L-LOC
کی (key)	O	O	O	O
پنجاب (Punjab)	I-ORG	B-ORG	B-ORG	B-ORG
یونیورسٹی (University)	I-ORG	I-ORG	L-ORG	L-ORG
میں (main)	O	O	O	O
کام (kam)	O	O	O	O
کیا (kiya)	O	O	O	O

The IO tagging scheme uses two tags, i.e., I (inside) and O (outside). If an NE, e.g., person name consists of one or more words, I-PER (inside) tag is assigned and O (other) tag is used for the remaining non-NE words. The BIO2 tagging scheme uses three tags to assign particular words. If an NE, e.g., person name is a single word then B-PER (Begin) tag is used. However, if an NE consists of two or more words then the B-PER tag is assigned to first word and the I-PER tag is assigned to all remaining words. The BILOU tagging scheme uses five tags. If an NE consists of a single word then the U-PER tag is used. If an NE consists of two words then the B-PER and L-PER tags are used for first and second words, respectively. If an NE consists of three or more words then the B-PER and L-PER tags are used for the first and last words, respectively, and the I-PER tag is used for all inside words. The BIL2 tagging scheme uses four tags. If an NE consists of a single word then L-PER tag is used. If an NE consists of two words then B-PER and L-PER are used. Finally, if an NE consists of more than

two words then for first word, last word, and all intermediate words are assigned B-PER, L-PER, and I-PER, respectively.

We used the following steps in our approach for Urdu NERC.

1) Selection of training and testing data.
2) Assignment of IO, BIO2, BILOU, and BIL2 tags to the training and testing data.
3) Build models using HMM [1] and CRF [5] for these tagging schemes.
4) Calculate F-measures using test data against the respective models.

The BIL2 tagging scheme produced better results than all other schemes for both machine learning algorithms. By using HMM, we used the trigram model with linear interpolation for smoothing. F-Measure for IO, BIO2, BILOU, and BIL2 were 44.87%, 44.88%, 45.14%, and 45.88%, respectively, In case of CRF, we used the base line model without using any features like neighboring words, prefixes, etc. F-Measure of IO, BIO2, BILOU, and BIL2 were 35.13%, 35.90%, 37.85% and 38.39%, respectively. There are chances that by using CRF with other features like previous words and Part Of Speech (POS), previous NE, we may achieve better results. Table 3 and Table 4 show the detailed results for HMM and CRF, respectively.

TABLE III. RESULTS OF FOUR TAGGING SCHEMES USING HMM

	Precision	Recall	F-Measure
IO	52.45	39.21	44.87
BIO2	54.04	38.38	44.88
BILOU	55.08	38.24	45.14
BIL2	55.22	39.24	45.88

TABLE IV. RESULTS OF FOUR TAGGING SCHEMES USING CRF

	Precision	Recall	F-Measure
IO	46.52	28.22	35.13
BIO2	47.34	28.91	35.90
BILOU	55.32	28.77	37.85
BIL2	55.57	29.32	38.39

Overall, the comparison of each tagging scheme with respect to HMM and CRF is shown in Figure 1. We use CRF without using any features like bigram, window size, and context. This is why the values of the performance measures for CRF for all tagging schemes are smaller than those of HMM. As you can see, by using IO tagging, HMM and CRF produced F-measure with least accuracies. Similarly, using BIO2 tagging, HMM and CRF produced F-measure better than IO but smaller than BILOU and BIL2. The same pattern can be observed in BILOU and BIL2 where using BILOU tagging HMM and CRF produced 2nd highest F-measure and using BIL2 both produced highest F-Measure as shown in Figure 1.

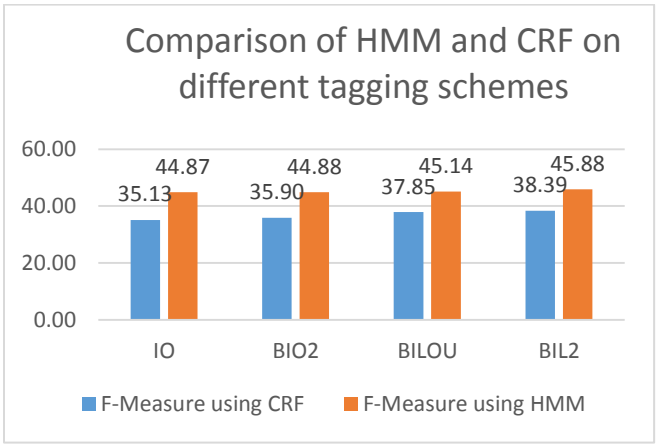

Fig. 1. Overall F-Measure results of CRF and HMM using each tagging scheme

The results of BIL2, IO, BILOU, and IOB2 using HMM on each NE and overall Precision, Recall, and F-Measure are shown in Figure 2, Figure 3, Figure 4, and Figure 5, respectively. With HMM, BIL2 and BILOU could not identify a single instance of NEA, NEB, NETE, NETO, and NETP, as shown in Figure 2 and Figure 4, respectively. NEB, NETE, NETO, and NETP could not be identified using HMM with IO and BIO2 tagging schemes, as shown in Figure 3 and Figure 5, respectively.

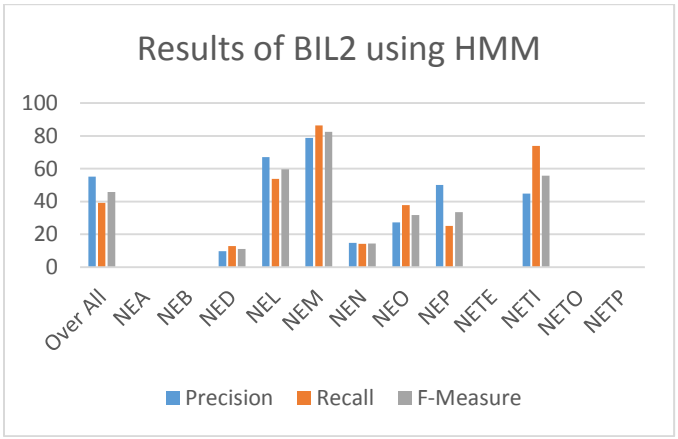

Fig. 2. F-Measure results of BIL2 tagging using HMM of each NE

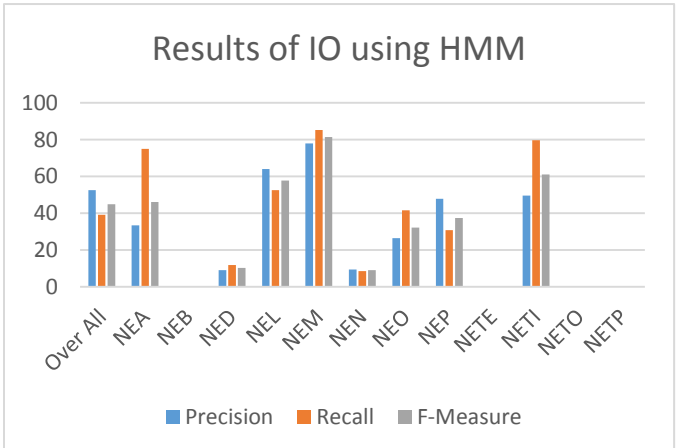

Fig. 3. F-Measure results of IO tagging using HMM of each NE

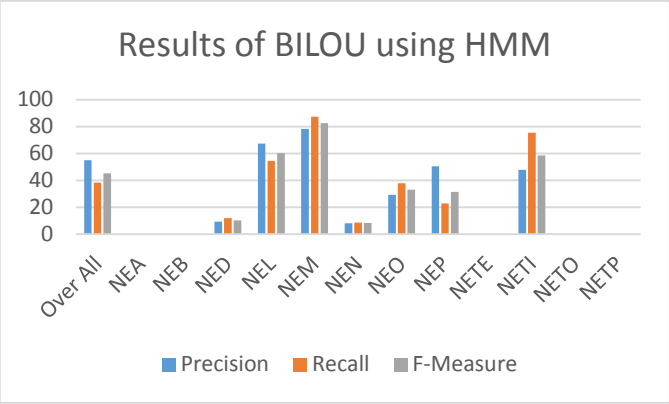

Fig. 4. F-Measure results of BILOU tagging using HMM of each NE

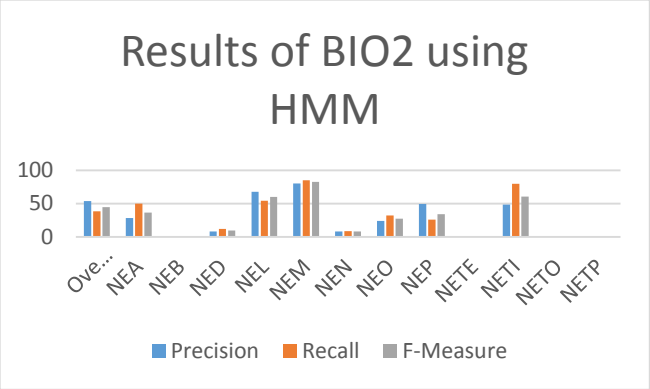

Fig. 5. F-Measure results of BIO2 tagging using HMM of each NE

Results of BIL2, IO, BILOU, and IOB2 using CRF on each NE and overall Precision, Recall, and F-Measure are shown in Figure 6, Figure 7, Figure 8, and Figure 9, respectively. Using CRF with BIL2 tagging did not identify a single instance of NEB, NETE, NETO, and NETP, as shown in Figure 6. NEB, NETE, NETO, and NETP could not be identified using CRF with IO tagging, as shown in Figure 7. The BILOU and BIO2 tagging schemes could not identify a single instance of NEA, NEB, NETE, NETO, and NETP, as shown in Figure 8 and Figure 9, respectively.

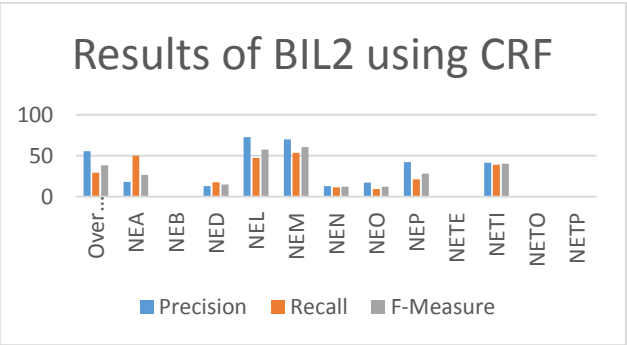

Fig. 6. F-Measure results of BIL2 tagging using CRF of each NE

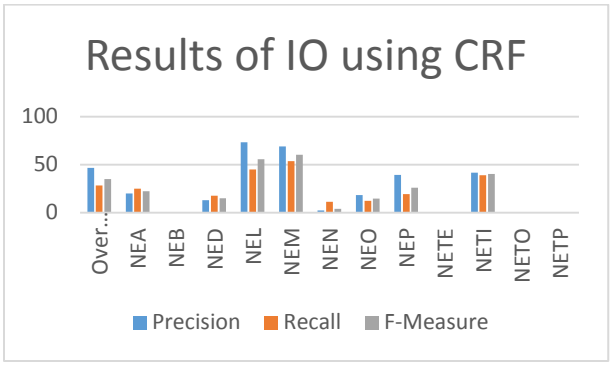

Fig. 7. F-Measure results of IO tagging using CRF of each NE

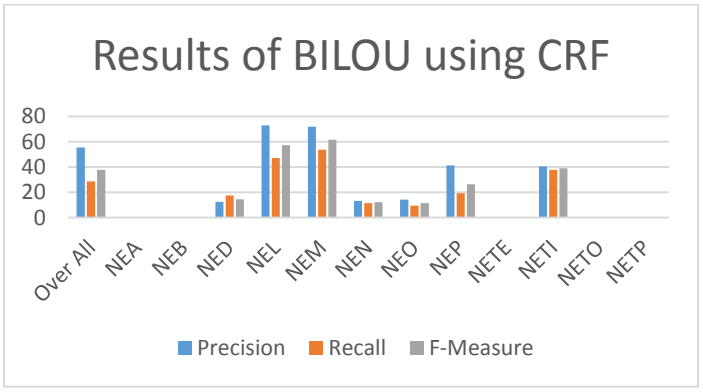

Fig. 8. F-Measure results of BILOU tagging using CRF of each NE

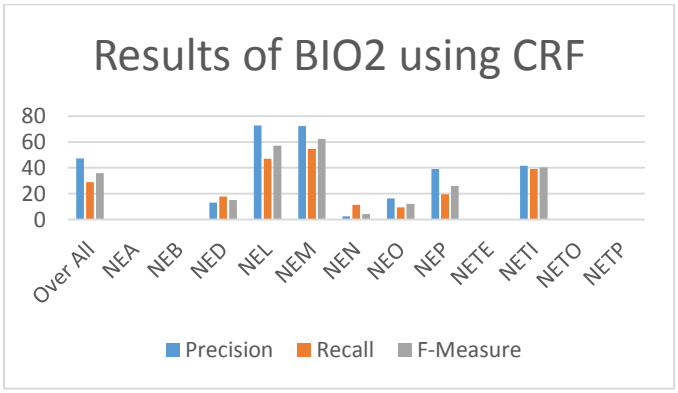

Fig. 9. F-Measure results of BIO2 tagging using CRF of each NE

F-Measure of each NE using HMM on each tagging scheme is shown in Figure 10. The figure shows that using HMM with any tagging scheme could not identify NEB, NETE, NETO, and NETP, and only IO and BIO2 tagging schemes identified NEA.

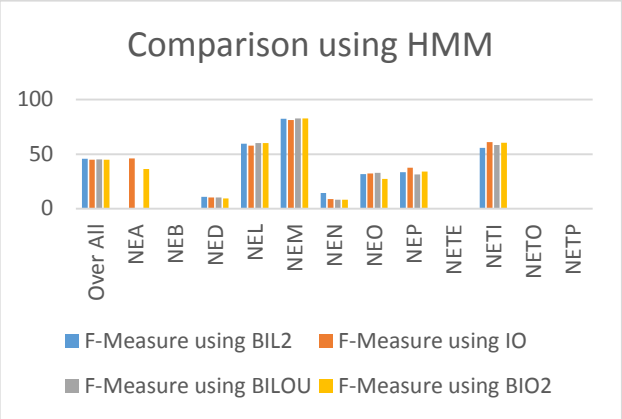

Fig. 10. F-Measure results of all tagging schemes using HMM of each NE

F-Measure of each NE using CRF with each tagging scheme is shown in Figure 11. The figure shows that using CRF with any tagging scheme could not identify NEB, NETE, NETO, and NETP, and only IO and BIL2 tagging schemes identified NEA. In summary, no tagging scheme with MMH or CRF could identify NEB, NETE, NETO, and NETP.

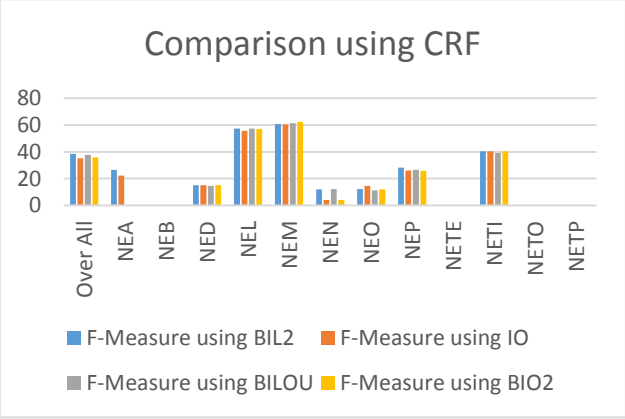

Fig. 11. F-Measure results of all tagging schemes using CRF of each NE

VII. Conclusion

The selection of an appropriate tagging scheme and selection of an appropriate ML algorithm may produce good results for the NER problem. In our experiments, we used a new NE tagging scheme for postpositional languages and compared the results with those obtained for the existing tagging schemes using HMM and CRF. The study shows that the NER tagging schemes for the Subject Verb Object (SVO) and SOV languages should be different for building a NER system with good F-measure values, because usually the SVO languages use the concept of preposition and the SOV languages use the concept of postpositional. Finally, our study shows that for Urdu, which is a postpositional language, the BIL2 tagging scheme generates the highest F-measure values using HMM and CRF.

VIII. Future Work

In future, we can perform experiments on other tagging schemes, including IOE and IOE2, to show a detailed comparison because these tagging schemes also support postposition languages. NER results with CRF using different features may be conducted to show that the BIL2 tagging scheme still performs better than others or not. Part of speech information can be used to improve the results of NER, and the list of person name, location name, and organization name may be exploited to improve results. We can also observe the improvement in the results of NER by using regular expressions for date, time, numbers, and measures.

References

[1] Brants, T. TnT: a statistical part-of-speech tagger. *In Proceedings of the sixth conference on Applied natural language processing* (pp. 224-231). Association for Computational Linguistics. (2000, April).

[2] Bikel, D. M., Miller, S., Schwartz, R., & Weischedel, R. Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing* (pp. 194-201). Association for Computational Linguistics. (1997, March).

[3] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, *12*, 2493-2537. (2011).

[4] Dai, H. J., Lai, P. T., Chang, Y. C., & Tsai, R. T. Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization. *Journal of Cheminformatics*, *7*(Suppl 1), S14. (2015).

[5] Finkel, J. R., Grenager, T., & Manning, C. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 363-370). Association for Computational Linguistics. (2005, June).

[6] Kudo, T., & Matsumoto, Y. Chunking with support vector machines. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies* (pp. 1-8). Association for Computational Linguistics. (2001, June).

[7] Krishnan, V., & Ganapathy, V. Named Entity Recognition. 2005.*Dostupno na: http://cs229. stanford. edu/proj2005/KrishnanGanapathy-NamedEntityRecognition. pdf (13.4. 2012)*. (2005).

[8] McCallum, A., & Li, W. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4* (pp. 188-191). Association for Computational Linguistics. (2003, May).

[9] Ramshaw, L. A., & Marcus, M. P. Text chunking using transformation-based learning. *arXiv preprint cmp-lg/9505040*. (1995).

[10] Ratinov, L., & Roth, D. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning* (pp. 147-155). Association for Computational Linguistics. (2009, June).

[11] Sang, E. F., & Veenstra, J. Representing text chunks. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics* (pp. 173-179). Association for Computational Linguistics. (1999, June).

[12] Tkachenko, A., Petmanson, T., & Laur, S. Named Entity Recognition in Estonian. *ACL 2013*, 78. (2013).

[13] Uchimoto, K., Ma, Q., Murata, M., Ozaku, H., & Isahara, H. Named entity extraction based on a maximum entropy model and transformation rules. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (pp. 326-335). Association for Computational Linguistics. (2000, October).