# N-gram and Gazetteer List Based Named Entity Recognition for Urdu: A Scarce Resourced Language

*Faryal Jahangir*[1], *Waqas Anwar*[1,2], *Usama Ijaz Bajwa*[1], *Xuan Wang*[2]
(1) COMSATS Institute of Information Technology, Abbottabad, University Road, Pakistan
(2)Harbin Institute of Technology Shenzhen Graduate School, P.R. China
faryaljahangir@ciit.net.pk, waqas@ciit.net.pk, usama@ciit.net.pk,
xuanwang@insun.hit.edu.cn

ABSTRACT

Extraction of named entities (NEs) from the text is an important operation in many natural language processing applications like information extraction, question answering, machine translation etc. Since early 1990s the researchers have taken greater interest in this field and a lot of work has been done regarding Named Entity Recognition (NER) in different languages of the world. Unfortunately Urdu language which is a scarce resourced language has not been taken into account. In this paper we present a statistical Named Entity Recognition (NER) system for Urdu language using two basic n-gram models, namely unigram and bigram. We have also made use of gazetteer lists with both techniques as well as some smoothing techniques with bigram NER tagger. This NER system is capable to recognize 5 classes of NEs using a training data containing 2313 NEs and test data containing 104 NEs. The unigram NER Tagger using gazetteer lists achieves up to 65.21% precision, 88.63% recall and 75.14% f-measure. While the bigram NER Tagger using gazetteer lists and Backoff smoothing achieves up to 66.20% precision, 88.18% recall and 75.83 f-measure.

KEYWORDS : Named Entity Recognition, Unigram model, Bigram model, Gazetteer lists, smoothing techniques

## 1 Introduction

Named Entity Recognition (NER) is a task that locates and classifies the named entities ('atomic elements') in a text into predefined classes/categories like the names of persons, organizations, locations, expressions of times, quantities, etc. For example consider the following sentence:

"Microsoft launched its first retail version of Microsoft Windows on November 20, 1985"

An accurate NER system would extract two NEs from the above sentence: (i) "Microsoft" as an organization and (ii) "November 20, 1985" as a date. The ambiguous nature of named entities makes the NER task very difficult and challenging, and because of this problem most of the NER systems fail to attain human level performance. NER is a basic tool for all application areas of Natural Language Processing (NLP) such as Automatic Summarization, Machine Translation, Information Extraction, Information Retrieval, Question Answering, Text Mining, Genetics etc. Performance of all these applications depends on the performance of the NER system. These applications can perform well if the named entities are recognized and grouped accurately.

This work presents a statistical approach using n-gram for Urdu NER. The objective of this NER system is to recognize five classes of NEs – Person, Location, Organization, Date and Time. In this

*Proceedings of the 10th Workshop on Asian Language Resources*, pages 95–104,
COLING 2012, Mumbai, December 2012.

work unigram and bigram models are used for NER and NE tagged training data is used to train these models. When these trained models are tested using test data, the results do not show a high recall because of the inherent problems of Urdu language like lack of resources and rich morphology. To improve the results of the statistical models, gazetteer lists are used. Due to the fact that very less research work has been done in Urdu language in the field of NLP, therefore standard sized corpus like Brown is not available for Urdu. We also used some smoothing methods with bigram model to solve sparse data problem. The smoothing techniques chosen to solve data sparseness are: Add-one, Lidstone, Witten-Bell and back-off. Among all of these smoothing techniques only the back-off technique has improved the results of Urdu bigram NER system.

Rest of the paper is organized as follows. A brief survey of different techniques used for the NER task in different languages is presented in Section 2. A discussion on the challenges for Urdu NER is given in Section 3. The proposed n-gram based NER system is described in Section 4. In Section 5 and 6 we present the experimental results and related discussions. Finally Section 7 concludes the paper.

## 2 Related work

A number of different techniques have been used for the development of NER systems for different languages since 1991. A surfeit of algorithms has been developed for NER of English and other European languages and has achieved high recognition rates. Comparatively very few NER algorithms have been developed for South and South East Asian languages especially for Urdu language. The following section discusses earlier research carried out to develop NER systems for different languages.

### 2.1 Rule based approaches

Among the earlier research papers in the field of NER area, (Lisa and Jacobs, 1991) has presented a rule based NER system for identification and classification of different company names. The accuracy of system is over 95%. (Cucerzan and Yarowsky, 1999) developed a language independent NER system for Hindi language by using contextual and morphological evidences for five languages such as English, Greek, Romanian, Turkish and Hindi. The performance of Hindi NER system is very low and has f-measure of 41.70 with very low 27.84% recall and nearly 85% precision.

### 2.2 Statistical approaches

(Bortwick, 1999) presented a NER system based on Maximum Entropy (ME) for English language and has achieved F-measure of 84.22%. (Li and MacCallum, 2003) presented a Conditional Random Field (CRF) for the development of NER system for Hindi language. The system has 71.50% accuracy. The authors provided large array of lexical test and used feature induction for constructing the features automatically. (Nadeau et al., 2006) presented semi-supervised approach for the development of English NER system by classifying 100 named entities. The System has achieved F-measure value in the range 78-87%. (Saha et al., 2008) have used Maximum Entropy based NER system for Hindi language. The system has achieved F-value of 80.01% by using word selection and word clustering based feature reduction techniques. (Ekbal and Bandyopadhyay, 2008) have developed a statistical Conditional Random Field (CRF) model for the development of NER system for South and South East Asian languages,

particularly for Bengali, Hindi, Telugu, Oriya and Urdu. The rules for identifying nested NEs for all the five languages and the gazetteer lists for Bengali and Hindi languages were used. The reported system achieved F-measure of 59.39% for Bengali,33.12 % for Hindi, 28.71% for Oriya, 4.749% for Telugu and 35.52 % for Urdu. (Goyal, 2008) developed CRF based NER system for Hindi language and evaluated it on test set1 and test set2 and achieved nested NEs F1-measure around 50.1% and maximal F1-measure around 49.2% for test set1 and nested NEs F1-measure around 43.70% and maximal F1 measure around 44.97 for test set2. (Gupta and Arora, 2009) presented a CRF based NER system for Hindi. The maximum F-measure achieved by the system is 66.7% for person, 69.5% for location and 58% for organization. (Raju et al. 2010) have developed ME based NER system for Telugu. The system has achieved an F-measure of 72.07% for person, 6.76%, 68.40% and 45.28% for organization, location and others respectively. (Ekbal and Saha et al., 2011) developed a multi-objective simulated annealing based classifier ensemble NER system for three scarce resourced languages like Hindi, Bengali and Telugu. The Recall, Precision and F-measure values are 93.95%, 95.15% and 94.55%, respectively for Bengali, 93.35%, 92.25% and 92.80%, respectively for Hindi and 84.02%, 96.56% and 89.85%, respectively for Telugu.

## 2.3 Hybrid approaches

(Bikel et al., 1997) developed IdentiFinder using HMM for English and Spanish languages to extract proper names and to make four categories including names, times, dates and numerical quantities. The system is reported to achieve F-measure of 90.44%. (Chaudhuri and Bhattachatya, 2008) developed NER system for Indian script Bangla. In which three-stage approach comprising of dictionary based, rules based and left-right co-occurrences statistics (n-gram) have been used for named entity. The system has achieved 85.50% recall, 94.24% precision and 89.51% f-measure. (Srikanth and Murthy, 2008) have used CRF based Noun Tagger for Telugu language using manually tagged data of 13,425 words for training and 6,223 words as test data. The system has F-value of Noun Tagger up to 92%. The rules based NER system has been developed for identifying names of person, place and organization. The overall F-measures of the system range between 80% to 97%. (Biswas et al., 2010) presented a hybrid system for Oriya NER based on ME, HMM and some handcrafted rules to recognize NEs. The system has an F-measure ranging between 75% to 90%. (Srivastava et al., 2011) presented hybrid approach for Hindi NER system. Rules were formulated over Conditional Random Field (CRF) model and Maximum Entropy (ME) model using features of POS and orthography for overcoming limitations of machine learning models for complex morphological languages like Hindi. The voting method has also been used to improve the performance of the NER system. Based on comparisons, CRF achieves better result than ME and rule based result.

## 2.4 Existing NE Systems for Urdu Language

Earlier research on NER for digital Urdu text has been carried out by (Becker and Riaz, 2002). Issues pertaining to Urdu language have been discussed and a corpus of 2200 Urdu documents has been developed. (Mukund et al., 2010) developed an information extraction system for Urdu language. The sub module of NER has been developed for information extraction system by using two models; namely ME and CRF based NER for Urdu. The result of ME has F-measures of 55.3% and the CRF based module for NER has F-measure value of 68.9%. (Riaz, 2010) has presented a rule based approach for Urdu NER system. Different rules have been formulated from 200 documents of Becker-Riaz corpus and have extracted 600 documents out of 2,262

documents for better evaluation during experimentation (Becker and Riaz, 2002). The system has f-measure of 91.1% with 90.7% recall and 91.5% precision. This rule based NER has achieved f-measures of 72.4% without any change in the rule set. The results have been later improved by developing new rules after analyzing the training set. The developed rule-based approach for Urdu NER shows encouraging results.

## 3 Challenges of Urdu NER

The large number of ambiguities of NE and the problems related to the Urdu language makes NER a challenging task. The construction of a robust Urdu NER is a complicated task because of the following limitations.

In English orthography capitalization of the initial letter is an indication that a word or sequence of words is a NE (Waqas et al., 2006). Urdu has no such indication which makes the detection of NEs more challenging. Thus, in Urdu language there is no difference between a NE and any other word from lexical point of view.

Some additional features can be added to the word to have more complex meaning. Agglutinative languages form sentences by adding a suffix to the root forms of the word. e.g. **پا کستا ن**(Pakistan is location) to **پا کستا نی** (Pakistani is person).

In Urdu Language SOV (Subject Object Verb) word order is used but usually the writers do not follow the same word order e.g**.** an English sentence "Ahmad closed the bag of books" can be written in Urdu**"کتابوں کا بستہ احمد نے بند کیا"** ("Kitabo ka basta Ahmad ne band kia") and **" احمد نے کتابوں کا بستہ بند کیا"**("Ahmad ne kitabo ka basta band kia"). The use of such different word orders makes the NE identification more challenging.

Some words are taken from other languages e.g. **(Palwasha) پلو شہ** is taken from Pushto language, **(Zeemal) ز یمل** is taken from Balochi language and **(Toyot) ٹویوٹا** is taken from English Language.

A nested name entity is composed of multiple words. This brings more challenges to accurately detect the beginning and the ending of a multi-word NE. To extract such NEs like **محمد علی جناح***(person name)*and**(Name (Organization) پشاور یونیورسٹی** as single NE is difficult. The NER system commonly extracts such NEs as separate NEs such as **پشاور** (location name) and **یونیورسٹی** (organization name).

Some entities are made up by using conjunction word such as **اور** e.g. **علی اوربلال سی این جی** (organization name) is a conjunct NE which cannot be recognized as a single NE by the NER system.

A name entity can be used as a person name or organization name or as a word other than nouns e.g. **نور** is a name of person and also equivalent to the English word "light".

## 4 Proposed n-gram based Urdu NER tagger

### 4.1 Unigram Model

Unigram model is the simplest form of n-gram models based on probability estimation approach. The unigram NE tagger assigns the most probable NE tags to the NEs. It is trained on the training data to calculate the probabilities of NEs. The most probable NE tag for a NE is

determined by calculating its probability with each NE tag. If the words in the corpus are given as $w_1, w_2, w_3, \ldots, w_n$ and their NE tags are represented as $t_1, t_2, t_3, \ldots..,t_n$. Then the unigram model calculates the maximum probability $P(t_i \mid w_i)$ and selects the most probable tag for each NE.Units.

### 4.2 Bigram Model

Bigram model is another form of n-gram model also based on probability estimation approach. The bigram NE tagger assigns the most probable NE tags to the NEs by considering the last encountered word i-e the bigram models looks one word back for probability estimation. The bigram model determines the most probable NE tag for a NE by calculating word and its tag probability with the previous word. The bigram model calculates the maximum probability $P(w_i t_i \mid w_{i-1})$ and selects the most probable tag for each NE.

### 4.3 Use of Gazetteer Lists

Due to the issues of Urdu language discussed in section 4 the statistical techniques could not show better results especially in case of recall rate. Due to wide variations and the agglutinative nature of South Asian Languages, probabilistic graphical models result into a low less recall rates. The gazetteer lists have been used in this work to improve the recall. As compared to other languages especially European languages, Urdu language processing is not mature yet so the language processing resources like gazetteer lists are not available. These gazetteer lists were prepared from different sources including internet. Lists for the following name entities were prepared: person names, location names, organization names, date, time. The data collected from the internet is not enough so the NE tagged corpus was also used to populate the gazetteer lists.

### 4.4 Use of Smoothing Techniques

The N-gram language models use Maximum Likelihood Estimation (MLE) for probability estimation. If the data occurs regularly in the training corpus the Maximum Likelihood Estimation (MLE) works better. The MLE uses counts of n-grams in training data; if N-gram has a zero count then its probability will also be zero which is called data sparseness. Data sparseness is the main problem for N-gram models especially when the available corpus is small sized. Due to insufficient amount of training corpus, the data sparseness problem is faced. To solve sparse data problem we have used different smoothing techniques as in (Daniel and James, 2009). Some of them have improved the results of n-gram model but others failed to improve the results. (Chen and Goodman, 1996) carried out an extensive empirical comparison of the most widely used smoothing techniques. Following smoothing techniques are used in this work. Add-one, Lidstone, Witten-Bell and Back-off smoothing techniques.

## 5 Experimental results and properties of the corpus

### 5.1 Propertied of the Training and Test Corpus

A NE tagged corpus has been downloaded from the CRL. 179896 tokens that have 938 NEs of this corpus are used to train the system and other 4917 tokens having 220 NEs are used as test corpus. The training corpus has been divided into four different sets to train the n-gram models. First we have taken Set1 and trained the n-gram models with it and obtained the test results. Then

we combine Set1 and Set2 to train the n-gram models and obtain the test results. After this we combine Set1, Set2 and Set3 to train the n-gram models and obtain the test results. At last we combine all the training data sets (Set1, Set2, Set3 and Set4) to train the n-gram models and obtain the test results. The testing data in all cases is same. The specification of these training data sets is given in the table 1 and the specification of testing corpus is given in table 2.

| Training sets | Total no. of tokens | Total no. of NEs | Total no. of NNEs |
|---|---|---|---|
| Set1 | 7972 | 367 | 7605 |
| Set2 | 8561 | 453 | 8108 |
| Set3 | 11500 | 555 | 10945 |
| Set4 | 17986 | 938 | 17048 |

TABLE 1- Specification of training corpus sets

Table 1 shows that set1 contains 7972 tokens; out of which there are 367 Named Entities and 7605 are not Named Entities.

| Total no. of tokens | Total no. of NEs | Total no. of NNEs |
|---|---|---|
| 4917 | 220 | 4697 |

TABLE 2- Specification of testing corpus

According to table 2 the testing corpus has 4917 tokens out of which 220 are NEs and the remaining 4697 tokens are not NEs. The tag set used is described in table 3

| Tag | Name | Description |
|---|---|---|
| </PERSON> | Person | عامر محمود، صدام<br>Sadaam ,Mehmood ,Amir |
| </LOCATION> | Location | پاکستان ، اسلام آباد، نئی دہلی<br>Pakistan, Islamabad, New Dehli |
| </ORGANIZATION> | Organization | مجلس عمل، لاہور ہائی کورٹ، |
| </DATE> | Date | پیر، جنوری، گیارہ ستمبر دو ہزار ایک |
| </TIME> | Time | نو بجے، شب، صبح |

TABLE 3- NE tag set

### 5.2 Evaluation Metrics

Before presenting the experimental results the evaluation parameters used for result's evaluation are discussed in this section. Massage Understanding Conference (MUC) and Multilingual Entity (MET) used the terms Precision (P) and Recall (R) from information retrieval research community which are now being used as evaluation metrics for performance of NER systems. Our NER system is evaluated in terms of precision, recall and f-measure.

### 5.3 Results of Unigram NER Tagger

The overall results of unigram NER Tagger with above specified training and testing data are given in Table4

| No. of Tokens/No. of NEs | Precision | Recall | F-measure |
|---|---|---|---|
| 7972/367 | 89.33 | 30.45 | 45.85 |
| 16533/820 | 89.53 | 35 | 50.33 |

| | | | |
|---|---|---|---|
| 28033/1375 | 88.46 | 41.81 | 56.79 |
| 46019/2313 | 85.71 | 49.09 | 59.09 |

TABLE 4- Results using simple unigram NER Tagger

The overall results by using unigram NER Tagger along with gazetteer lists are given in table 5.

| No. of Tokens/No. of NEs | Precision | Recall | F-measure |
|---|---|---|---|
| 7972/367 | 65.52 | 87.27 | 74.85 |
| 16533/820 | 65.87 | 88.63 | 75.58 |
| 28033/1375 | 65.99 | 89.09 | 75.82 |
| 46019/2313 | 65.21 | 88.63 | 75.14 |

TABLE 5-Results using unigram NER Tagger along with gazetteer lists

The results we obtained for different types of NEs using unigram NER Tagger along with gazetteer lists are given in table 6.

| Types of NEs | Precision | Recall | F-measure |
|---|---|---|---|
| Location | 85.04 | 94.79 | 89.65 |
| Person | 48.734 | 90.58 | 63.37 |
| Organization | 80 | 44.44 | 57.14 |
| Time | 66.66 | 100 | 80.00 |
| Date | 87.5 | 63.63 | 73.68 |

TABLE 6 -Results using unigram NER Tagger along with gazetteer lists for different types of NEs

### 5.4 Results of Bigram NER Tagger

The overall results of the simple bigram NER Tagger are given in Table 7

| No. of Tokens/No. of NEs | Precision | Recall | F-Measure |
|---|---|---|---|
| 7972/367 | 90.91 | 9.09 | 16.5 |
| 16533/820 | 88.89 | 10.91 | 19.44 |
| 28033/1375 | 92.31 | 16.37 | 27.78 |
| 46019/2313 | 88 | 20 | 32.59 |

TABLE 7- Overall results using bigram NER Tagger

The overall results obtained after applying gazetteer lists to the tagged data returned by bigram NER tagger are given in Table 8.

| No. of Tokens/No. of NEs | Precision | Recall | F-Measure |
|---|---|---|---|
| 7972/367 | 65.26 | 84.54 | 73.66 |
| 16533/820 | 65.38 | 85 | 73.91 |
| 28033/1375 | 65.38 | 85 | 73.91 |
| 46019/2313 | 64.58 | 84.54 | 73.23 |

TABLE 8- Results using bigram NER Tagger along with gazetteer lists

The overall results by using bigram NER Tagger along with gazetteer lists and Backoff Smoothing are given in table 9.

| No. of Tokens/No. of NEs | Precision | Recall | F-Measure |
|---|---|---|---|
| 7972/367 | 65.39 | 85.90 | 74.26 |
| 16533/820 | 65.8 | 87.72 | 75.24 |
| 28033/1375 | 66.10 | 88.63 | 75.72 |
| 46019/2313 | 66.20 | 88.18 | 75.83 |

Table 9 Overall results using bigram NER Tagger along with gazetteer lists and Backoff Smoothing

The results we obtained for different types of NEs using bigram NER Tagger along with gazetteer lists and Backoff Smoothing are given in table 10.

| Types of NEs | Precision | Recall | F-measure |
|---|---|---|---|
| Location | 84.07 | 98.95 | 90.90 |
| Person | 49.04 | 90.58 | 63.63 |
| Organization | 93.33 | 38.88 | 54.90 |
| Time | 100 | 50 | 66.66 |
| Date | 87.67 | 63.63 | 73.75 |

TABLE 10-Results using bigram NER Tagger along with gazetteer lists and Backoff smoothing for different types of NEs

## 6 Discussion

From Table 4 and 7 we can see that simple unigram and bigram models produce a high precision but the recall is very low in both cases because of small sized training data. To improve our recall we used gazetteer lists along with unigram and bigram Taggers. By using the gazetteer list the recall of the taggers improved but at the cost of precision. Here the precision decreases because our tagger looks the gazetteers one by one in a sequence and tags a word with respect to the type of the list in which it finds the words first without any confirmation whether it's the right tag for that word or not. Resultantly it tags many words incorrectly which decreases the recall. As compared to unigram Tagger, bigram Tagger show very low recall, because the bigram NER Tagger uses word bigram for probability calculation so it needs more training data as compared to unigram NER Tagger. Since we have a small sized training corpus, so the bigram NER Tagger finds only some of the NE bigrams in training corpus and tags them with appropriate tags and misses a large number of NEs due to data sparseness. To solve this sparse data problem some smoothing techniques were tested with bigram model and among all the techniques tested, only back off smoothing improved the results. From the results, it is evident that as the size of training data increased the results of the taggers got better. But in case of training Set4, the results especially recall decreased, because the NEs present in training Set4 create more ambiguity, as they belong to more than one type of NE classes depending on the context in which they are used.

## Conclusion

In this research work we presented a statistical NER tagger for Urdu language. There are various issues related to Urdu language processing, including lack of standard Urdu corpus and incompatibility issues of NLP tools for Urdu language which has been discussed earlier. In this work NER for Urdu text has been implemented using unigram and bigram statistical models. Significant results have been produced even with a small sized training data. Low recall and sparse data problems occur due to the inherent issues of Urdu language like unavailability of sufficient

resources. To solve sparse data problem we tested different smoothing techniques and the backoff smoothing technique proved beneficial. We also used gazetteer lists to improve the results of n-gram statistical models. The unigram tagger trained with training data and combined with gazetteers produced up to 65.217% precision, 88.636% recall and 75.144% f-measure. A bigram NER tagger is trained with training data, combined with gazetteers and Backoff smoothing produced up to 66.205% precision, 88.181% recall and 75.834% f-measure.

## References


D. Becker, K. Riaz. (2002). A study in urdu corpus construction. Proceedings *of the 3rd Workshop on Asian Language Resources and International Standardization*, pages 1–5.

DM. Bikel, S. Miller, R. Schwartz, , R. Weischedel. (1997). Nymble: a high-performance learning name-finder. *In Proceedings of the fifth Conference on Applied Natural Language Processing*, pages 194-201.

S. Biswas, S. Mishra, S. Acharya, S. Mohanty. (2010). A hybrid oriya named entity recognition system: harnessing the power of rule Biswas". *International Journal of Artificial Intelligence and Expert Systems*. 1(1): 1-6.

A. Borthwick. (1999). A maximum entropy approach to named entity recognition. New York University.

BB. Chaudhuri, S. Bhattacharya. (2008). An experiment on automatic detection of named entities in Bangla. *In Proceedings of the IJCNLP Workshop on NER for South and South East Asian Languages*, pages 75-81.

SF. Chen, J. Goodman. (1996). An empirical study of smoothing techniques for language modeling. *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310-318.

S. Cucerzan, D. Yarowsky. (1999). Language independent named entity recognition combining morphological and contextual evidence. *Proceedings of the Joint SIGDAT conference on EMNLP and VLC,* pages 90-99.

J. Daniel, HM. James. (2009). *Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition* (2nd.). Prentice Hall.

Ekbal A, Bandyopadhyay S. (2008). Bengali named entity recognition using conditional random field. *In Proceedings of the IJCNLP Workshop on NER for South and South East Asian Languages,* pages 589-594.

A. Ekbal, S. Saha. (2011). A multiobjective simulated annealing approach for classifier ensemble: Named entity recognition in Indian languages as case studies. Expert *Systems with Applications*. 38(12): 14760-14772.

Goyal. (2008). Named entity recognition for SouthAsian languages. *Proceedings of the IJCNLP Workshop on NER for South and South East Asian Languages*, pages 89-96.

Gupta PK, Arora S. (2009). An approach for named entity recognition system for Hindi: an experimental study. *In Proceedings of the ASCNT,* pages 103 –108.

Li W, McCallum A. (2003). Rapid development of Hindi named entity recognition using conditional random fields and feature induction. *In ACM Transactions on Asian Language Information Processing*. 2(3): 290-294.

Lisa FR, Jacobs SP. (1991). Creating segmented databases from free text for text retrieval. *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval,* pages 337--346.

Mukund S, Srihari R, Peterson E. (2010). An Information-extraction system for Urdu—a resource-poor language. *In ACM Transactions on Asian Language Information Processing*. 9(4):15.

Nadeau D, Turney P, Matwin S (2006). Unsupervised named-entity recognition: generating gazetteers and resolving ambiguity. *Canadian Conference on Artificial Intelligence*, pages 266-277.

Raju SB, Raju DSV, Kumar, K (2010). Named entity recognition for Telegu using maximum entropy model. *Journal of Theoretical and Applied Information Technology*. 13(2): 125-130

Riaz K. (2010). Rule-based named entity recognition in Urdu. *In Proceedings of the Named Entities Workshop*. pages 126-135.

Saha SK, Sudeshna S, Mitra P. (2008). A hybrid feature set based maximum entropy Hindi named entity recognition. *Proceedings of the Third International Joint Conference on Natural Language Processing*, Pages 343-349

Srikanth P, Murthy KN. (2008). Named entity recognition for Telugu. *Proceedings of the IJCNLP Workshop on NER for South and South East Asian Languages*, pages 41-50.

Srivastava S, Sanglikar M, Kothari D. (2011). Named entity recognition system for Hindi language: a hybrid approach. *International Journal of Computational Linguistics*. 2(1): 10-23.

Waqas A, Xuan W, Xiao-long W. (2006). A Survey of Automatic Urdu language processing. *International Conference on Machine Learning and Cybernetics*, pages 4489-4494.