Rule-based Named Entity Recognition in Urdu

Kashif Riaz
University of Minnesota
Department of Computer Science
Minneapolis, MN, USA
riaz@cs.umn.edu

Abstract

Named Entity Recognition or Extraction (NER) is an important task for automated text processing for industries and academia engaged in the field of language processing, intelligence gathering and Bioinformatics. In this paper we discuss the general problem of Named Entity Recognition, more specifically the challenges in NER in languages that do not have language resources e.g. large annotated corpora. We specifically address the challenges for Urdu NER and differentiate it from other South Asian (Indic) languages. We discuss the differences between Hindi and Urdu and conclude that the NER computational models for Hindi cannot be applied to Urdu. A rule-based Urdu NER algorithm is presented that outperforms the models that use statistical learning.

1. Introduction

Text processing applications, such as machine translation, information extraction, information retrieval or natural language understanding systems need to recognize multiple word expressions that refer to people names, organizational names, geographical locations, and other named entities. Proper Names play a crucial role in information management, both in specific applications and in underlying technologies that drive the application. Name Recognition becomes important in situations when the person or the organization is more important than the action it performed, for example, bankruptcy of the corner shop John & Sons is not as interesting as the bankruptcy of General Motors, an American car manufacturer. In this particular example, latter event will be of much interest for the financial markets and investors to track.

The proper name identification depends upon the domain, and the applications in that domain. For the purpose of this study we have limited the scope of names to entities proposed by Palmer and Day (1996), i.e. times, numbers, personal names, organizations, and geographical areas. The goal of a named entity finder is to find these entities.

In this paper we study the challenges of named entity recognition for resource scarce languages among South Asian languages. Urdu is used as an example language because of its large number of speakers, the only language in the region with Arabic script orthography, and interesting assumptions about its similarity with Hindi. Section 2 describes the characteristics and computational processing for Urdu. Section 3 motivates the named entity recognition task by outlining the challenges in NER in any language along with some of the approaches that have been used by well known NER systems. Section 4 discusses some previous work related to NER in South Asian languages. Section 5 describes challenges of NER in Urdu. Section 6 describes the complex relationship between Hindi and Urdu and asserts that NER computation models for Hindi cannot be used for Urdu NER. Section 7 presents a rule-based NER algorithm for Urdu NER. Section 8 presents the conclusion and future work. It is assumed that the reader knows the history, orthography and some characteristics of Urdu in general. We give a brief introduction to Urdu and Urdu processing in section 1.1. For a detailed explanation refer to Riaz (2008) that describe computational challenges for Urdu processing.

As a convention, Urdu words written in Arabic orthography are followed by English translation in parenthesis and are italicized.

2. Characteristics of Urdu

This section briefly introduces some right to left languages and a few characteristics of Urdu. Urdu is the national language of Pakistan, and one of the major languages of India. It is estimated that there are about 300 million speakers of Urdu. Most of the Urdu speakers live in Pakistan, India, UAE, U.K and USA. Recently, there has been a lot of interest in

Proceedings of the 2010 Named Entities Workshop, ACL 2010, pages 126–135,
Uppsala, Sweden, 16 July 2010. ©2010 Association for Computational Linguistics

computational processing of right to left languages. Most of the interest has been focused toward Arabic. There are other right to left languages like Urdu, Persian (Farsi), Dari, Punjabi, and Pashto that are mostly spoken in South Asia. Arabic is a Semitic language and the other languages belong to the Proto Indo Iranian languages. Arabic and these other languages only share script and some vocabulary. Therefore, the language specific task done for Arabic is not applicable to these languages. For example, stemming algorithms generated for Arabic will not work for a language like Urdu.

Unlike other languages in South Asia, Urdu shares its grammar with Hindi. The difference is vocabulary, and writing style. Hindi is written in Devanagri script whereas Urdu is written in Arabic script. Because of these similarities, Hindi and Urdu are considered one language for linguistic purposes but current Hindi resources cannot be used for Urdu processing (Riaz, 2009). Urdu is quite complex language because Urdu's grammar and morphology is a combination of many languages: Sanskrit, Arabic, Farsi, English and Turkish to name a few. Urdu's descriptive power is quite high. This means that there could be many different ways in which a concept can be expressed in Urdu. For example, in Urdu the words *Pachem* and *Maghreb* both are used for the direction *West.* In the previous example *Pachem* has its ancestry in Sanskrit and *Maghreb* has its roots in Arabic. Urdu is considered the *lingua franca* of business in Pakistan, and the South Asian community in the U.K (Baker et. al, 2003).

Urdu has a property of accepting lexical features and vocabulary from other languages, most notably English. This is called code-switching in linguistics e.g. it is not uncommon to see a right to left flow interrupted by a word written in English (left to right) and then continuation of the flow right to left. For example,وہ میرا laptop ہے *(That is my laptop*). In the above example, Microsoft Word did not support English embedding within the Urdu sentence and displayed it improperly. But while electronically processing, the tokenization will be done correctly (Becker and Riaz, 2002). In order to process Urdu and other right to left languages Unicode encoding and proper font usage is necessary. Becker and Riaz (2002) discuss Urdu Unicode encoding in detail.

3. Challenges in NER

Named Entity Recognition was first introduced as part of Message Understanding Conference (MUC-6) in 1995 and a related conference MET-1 in 1996 introduced named entity recognition in non-English text. In spite of the recognized importance of names in applications, most text processing applications such as search systems, spelling checkers, and document management systems, do not treat proper names correctly. This suggests proper names are difficult to identify and interpret in unstructured text. Generally, names can have innumerable structure in and across languages. Names can overlap with other names and other words. Simple clues like capitalization can be misleading for English and mostly not present in non western languages like Urdu.

The goal of NER is first to recognize the potential named entities and then resolve the ambiguity in the name. There are two types of ambiguities in names, structural ambiguity and semantic ambiguity. Wacholder et al. (1997) describes these ambiguities in detail. Non-English names pose another dimension of problems in NER e.g. the most common first name in the world is Muhammad, which can be transliterated as Mohmmed, Muhammad, Mohammad, Mohamed, Mohd and many other variations. These variations make it difficult to find the intended named entity. This transliteration problem can be solved if the name Muhammad is written in Arabic script as محمد.

3.1 General Approaches to NER

Over the years many systems have been crafted to find names in different domains. Some are quite general and work in all domains, while others are domain specific. The domain specific systems do much better in their domains and perform poorly on foreign domains. On the other hand the systems that claim generality do not work as well as the best domain specific systems but do not fare poorly when the domain is changed.

Nymble (Bikel et al, 1996) is a purely statistical model where named entities are found using a generative statistical model using a variant of HMM (Hidden Markov Model). Recently, statistical discriminative models like Condition Random Fields (CRF) (Wallah, 2002) are used consistently for segmenting and labeling the sequence data as a graphical model (Lafferty et al. 2009). Nominator (Wacholder et al, 1997) is a fully implemented module for proper name recognition. It applies a set of heuristics to a list of words based on patterns of capitalization, punctuation and location within the sentence. Dr. Hermansen at Linguistic Analysis Systems Inc. has a well known system that recognizes names based on regional names (Erickson, 2005).

4. NER for South Asian languages and Related Work

Although over the years there has been considerable work done for NER in English and other European languages, the interest in the South Asian languages has been quite low until recently. One of the major reasons for the lack of research is the lack of enabling technologies like, parts of speech taggers, gazetteers, and most importantly, corpora and annotated training and test sets. One of the first NER study of South Asian languages and specifically on Urdu was done by Becker and Riaz (2002) who studied the challenges of NER in Urdu text without any available resources at the time. The by-product of that study was the creation of Becker-Riaz Urdu Corpus (2002). Another notable example of NER in South Asian language is DARPA's TIDES surprise language challenge where a new language is announced by the agency to build language processing tools in a short period of time. In 2003 the language chosen was Hindi. Li and McCallum (2003) tried conditional random fields on Hindi data and reported *f-measure* ranging from 56 to 71 with different boosting methods. Mukund et al. (2009) used CRF for Urdu NER and showed *f-measure* of 68.9%.

By far the most comprehensive attempt made to study NER for South Asian and South East Asian languages was by the NER workshop of International Joint Conference of Natural Language Processing in 2008. The workshop attempted to do Named Entity Recognition in Hindi, Bengali, Telugu, Oriya, and Urdu. Among all these languages Urdu is the only one that has Arabic script. Test and training data was provided for each language by different organizations therefore the quantity of the annotated data varied among different languages. Hindi and Bengali led the way with the most amounts of data; Urdu and Oriya were at the bottom with the least amount of data. Urdu had about 36,000 thousand tokens available. A shared task was defined to find named entities in the languages chosen by the researcher. There are 15 papers in the final proceedings of NER workshop at IJCNLP 2008, all cited in the references section, a significant number of those papers tried to address all languages in general, but resorted to Hindi, where the most number of resources were available. Some papers only addressed specific languages like Hindi, Bengali, Telugu and one paper addressed Tamil. There was not a single paper that focused on only Urdu named entity recognition. The papers that tried to address all languages, the computational model showed the lowest performance on Urdu. Among the experiments performed at Named Entity Workshop on various Indic languages and Urdu, almost all experiments used CFR with limited success.

4. NER challenges for Urdu

In general NER is a difficult task and a number of challenges need to be addressed in all languages. South Asian languages have some additional challenges. We will focus on language characteristics and some practical problems of language processing focusing on Urdu for examples. It is important to note that the following characteristics are not unique to Urdu nor to the South Asian languages.

5.1 No Capitalization

Capitalization, when available, is the most important feature for named entity extraction. English and many other European languages use it to recognize proper names. Orthography of Urdu does not support capitalization. English systems easily recognize acronyms by using capitalization, but in Urdu they are quite difficult to recognize. For example, بی بی سی *(transcribed BBC)* in Urdu cannot be recognized as an acronym.

5.2 Agglutinative nature

Agglutinative property means that some additional features can be added to the word to add more complex meaning. Agglutinative languages form sentences by adding a suffix to the root forms of the word. This feature was mentioned in relation to Telugu only in the NER literature of IJCNLP 2008 presuming unfamiliarity to Urdu by the authors. A deeper study shows that agglutinative nature of Urdu comes from Persian, Turkish and Dravidian languages. In Urdu *Hyderabad* + *i* = *Hyderabadi* حیدرآبادی; the root word is *Hyderabad* and the suffix is *i*. Here *Hyderabadi* should not be recognized as a named entity whereas *Hyderabad (city in India)* should be recognized as a location named entity.

5.3 Ambiguity

Ambiguity in proper name names is present in South Asian languages as in English. The names like Brown are ambiguous in English – name or color. Similarly, سحر *(Sahar)* is ambiguous in Urdu – name or morning dawn. In Urdu this gets more complicated because سحر *(Sahar)* also means a spell.

Common nouns can be used as proper names in South Asian languages. An example in Urdu is کریم *(generosity)* which is also a man's name.

5.4 Word Order

A number of South Asian languages have a different word-order than English and some have a free word-order. Urdu mostly has a word order but depending upon the domain the word order is not respected. e.g. *Jamal ne paani ka pura glass piya* and *Panni ka glass Jamal ne pura piya* both translates to *Jamal drank a whole glass of water.*

5.5 Spelling Variations

A number of situations occur in news articles where different authors or reporters scribe the name in different spellings even for native Urdu names. In English, this is recognized by capitalization and but in Urdu in the absence of capitalization this becomes a problem. An example is مسعود and مسود, where both strings represent the same person Masood. مسعود *(Masood)* represents the Arabic style of writing the name with an extra vowel and مسود *(Masood)* is written in the native Urdu form.

5.6 Ambiguity in Suffixes

A very common phenomenon in the proper names and common name in the South Asian languages is the use of a location suffix in a name. Sometimes the suffix is attached to the location name like a building or a road. A common practice is to append the location of person's origin in a name with a suffix *-i* or *-vi*. For example, if a person was from Batala (city in the Indian Punjab), *-vi* is added to the name to form *Batalvi*. This is observed in Urdu because most poets of Urdu use a name of their choosing, like an alias, at the end their name. This alias is called *takhalus* to refer themselves in their poetry. Almost always these names in absence of the poetic context are meaningful words that are not named entities.

5.7 Loan words in Urdu

Urdu has a number of loan words. Loan words are words that are not indigenous to Urdu. The named entity recognizer that is based on simple morphological cues will fail to recognize a large number of proper nouns. For example, گوانتاناموبے (*Guantanamo Bay)* is an English word with *Bay* as a cue for location. Similarly, for Osama Bin Laden, بن *(bin)* an Arabic cue needs to be used in the middle of the name for the person name.

5.8 Nested Entities

The named entities that are classified as nested contain two proper names that are nested together to form a new named entity. An example in Urdu is *Punjab University* where *Punjab* is the location name and *University* marks the whole entity as an organization.

5.9 Conjunction Ambiguity

Urdu text shows quite a few examples of conjunction ambiguities among proper nouns. That is, there is an ambiguity if the entity is one proper noun or two proper nouns e.g. *Toyota and Honda motor company* in English. Although, this phenomenon is present in most languages none of the papers in IJCNLP NER workshop mentioned them as a problem. An example of conjunction ambiguity is گوگل اور یاھو نے کھانا دیا *(Google and Yahoo offered banquet).*

5.10 Resource Challenges

NER approaches are either based on rule engine or inference engines. In each approach some type of corpus is required; lack of a large corpus for deriving rules is an issue for most South Asian languages, Urdu in particular. There are only two corpora available EMILLE corpus (Baker, et al., 2003) and Becker-Riaz (2002) corpus. The EMILLE corpus contains long running articles that do not have a lot of named entities. Becker-Riaz corpus contains short news articles and has a very rich content for named entity recognition. NER workshop at IJCNLP 2008 did not use either of them and contained only 36,000 Urdu tokens.

Recent experiments in NER in almost all aspects have been conducted through the use of inference engines using statistical machine learning. In the NER workshop at IJCNLP 2008, with one exception, all experiments used statistical machine learning for name recognition and conditional random fields (CRF) was favored by the majority. A good large annotated corpus is the pre-requisite to learn the rules. All experiments that used pure machine learning performed poorly and had to boost the performance of the system using gazetteers, online dictionaries and other hand crafted rules.

Urdu NER performed poorly and mostly at the bottom for each experiment and all researchers claimed the lack of the other resources to boost its performance. In summary, there is a dearth of annotated corpus for named entities for NER for South Asian languages. Urdu and Oriya are two languages where researchers could not find any gazetteers and online dictionaries for boosting the performance of the algorithms.

6. Analysis of Urdu and Hindi

Since Hindi NER was satisfactory in NER workshop at IJCNLP 2008 and Urdu and Hindi are closely related languages, a claim can be made that any computational model or algorithm that works for Hindi should work for Urdu also. This section describes in detail that this assertion is invalid for computational processing and sharing of resources. Extensive research has been done about the ancestry of Urdu and Hindi and their origins but no research study exists that compares and contrasts Urdu and Hindi in a scholarly fashion (Russell, 1996). Some rudimentary experiments for computationally recognizing names show that Hindi and Urdu behaved as two different languages. For example, while trying to recognize the capitol, the cues of recognition of locations are different e.g. *Dar-al-Khilafah* (Urdu) and *Rajdihani* (Hindi) are both used for the capitol of a city or a country. Therefore, we concluded that more research is warranted to understand the relationship between these two languages to understand if the computational models based one language can be used in some capacity for the other language.

The relationship between Hindi and Urdu is very complex, while analyzing the differences at high level they can be treated as the same language and play pivotal role in establishing the links between other South Asian communities across the world. At detailed levels they are separate languages and deserve to be studied and treated as separate languages. This is most apparent in the official documents produced by the Indian government in Hindi and news broadcasts that are not understandable by Urdu speakers (Matthews, 2002). The following example is borrowed from Russell (1996) to explain the growing divergence between Hindi and Urdu. Consider the sentence in English "*The eighteenth century was the period of the social, economic and political decline*". The Urdu translation of the sentence is "*Atharvin sadi samaji, iqtisadi aur siyasi zaval ka daur tha*" while the Hindi equivalent is "*Atharvin sadi samajik, arthik aur rajnitik girav ki sadi thi*". Russell points out that this example shows alone that Urdu speakers cannot understand the meaning of the Hindi equivalent and vice versa. Therefore, these two languages should not be treated as the same language in all circumstances.

We assert that that computational models built for one of the languages cannot be translated for the other language. A case in point is Hindi Wordnet (Jha et al., 2001), which is an excellent source for Hindi language processing but cannot be used for Urdu, because of the explanations given earlier. In addition, the following properties of the Hindi Wordnet make it unusable for Urdu processing without extra ordinary amount of work: The terminology used to describe parts of speech (POS) in Hindi Wordnet is completely foreign to Urdu speaker. Also, the POS names are Sanskrit-based whereas the Urdu POS are Persian and Arabic based. For example, in Hindi the word for noun is *sangya* and in Urdu it is called *ism.* The proper noun in Hindi is called *vyakti vachak sangy*, no Urdu speaker will know this unless they have studied Hindi grammar. In order to work through these differences, one has to be familiar with both languages at almost expert levels. In other words in order to use Hindi resources to do Urdu computational processing one has to know Hindi at detailed linguistic level. A detailed analysis of phonological differences between Hind and Urdu and the resource construction of Hindi using Highbrow formalisms is discussed in detail by Riaz (2009).

7. Rule-based Urdu NER

We used a hand crafted rule-based NER system for Urdu NER instead of using a machine learning approach for the following reasons:

- There are no good annotated corpora available. The only annotated corpus available is through the NER workshop of IJCNLP 2008 which is only 36000 words.
- At NER workshop IJCNLP 2008 Urdu data was available to all the researchers but none of the experiment fared well for Urdu using CRF.
- Conditional Random Fields (CRF) is the state of art for named entity extraction, in the absence of boosting methods like gazetteers, CRF performed poorly with only annotated text.

- There are no gazetteers and online dictionaries available for Urdu that are accessible through Web Services or for online consumption.
- Hindi resources cannot be used to bridge the lack of language resources for Urdu (Riaz, 2009).
- Creating a new set of tagged data set for modeling CRF or other new statistical algorithm on Urdu data is cost prohibitive at this time.

7.1 Experiment Setup

There are two corpora available for Urdu for research in NER; Becker-Riaz corpus and EMILLE corpus. Although EMILLE is a larger corpus, it contains articles that are long and deficient of named entities. Becker-Riaz corpus is a news article corpus and it contains abundant of named entities. We chose 2,262 documents from the Becker-Riaz corpus and removed a number of XML tags and their content for readability. A sample document from the reduced Becker-Riaz corpus is constructed by using XSLT is given below:

```
 <cesDoc>
<doc-number>021003_uschinairaq_atif</doc-number>
<title>نئی عراق:نئی قرارداد روس کو قبول </title>
<para>امریکی ایوان
     نمائندگان نے بدھ کو عراق سے متعلق صدر بش کی پالیسی کی سیاسی
     حمایت کی ہے جس کے باعث بظاہر امریکہ کے لئے بغداد کے خلاف
     عسکری قوت
     استعمال کرنے کی راہ ہموار ہو جائے گی۔ تاہم امریکی سینیٹ اب اس
     کرنے کی راہ ہموار ہو جائے گی۔ تاہم امریکی سینیٹ اب  معاملے پر غور
     اس معاملے
          کرے گی۔</para>
</cesDoc>
```

The documents are not tagged with named entities so rules need to be constructed to find proper names. A number of proper noun cues are available in the text to generate those rules. About 200 documents were analyzed to construct the set of rules, while analyzing text a number of ambiguities were found – some of those are discussed in the earlier sections. The rules were constructed for the following named entities – examples are given in English for clarity.

- Person name e.g. George Bush
- Person of influence if proper name is identified e.g. President George Bush
- Location name e.g. Pakistan, Bharat, Punjab, America, Lahore
- Date: 1996
- Numbers: e.g. 31,000
- Organization e.g. Taliban, Al-Qaeda, B.B.C.

Although rules are designed to recognize the above named entities, the current implementation recognizes all of them as simple named-entities. While crafting rules for named entities a number of interesting rule patterns, heuristics and challenges were discovered that play important role when discovering a named entity. We mention some interesting ones below:

- Punctuation marks like ":" are useful but the position of their occurrence in text is important.
- Beginning of the sentence in title of news text has a different rule than beginning of the sentence in the paragraph text.
- Titles of the news text are not grammatically formed. A rudimentary POS tagger available from CRULP (Center for Research in Urdu language Processing) fails on marking the constituents of sentence. Moreover, POS tagger changed the order of words. This further complicated writing matching rules.
- Stemming reduces the precision of the system. It will conflate terms like *Pakistani* to *Pakistan.* Hence, marking *Pakistan* as named entity in the context of the *Pakistani* which is not a named entity.
- Suffix rules are very helpful in recognition of location names e.g. *–stan* for Pakistan, Afghanistan etc. But it does not find names like *Bharat, Iran* etc.
- Same suffix can identify location and organization e.g. *Taliban* and *Afghanistan.*
- String of names like *Rahid Latif, Shahid Afridi, and Muhammad Yousaf* are problematic for our NER system since there is no capitalization in Urdu and they occur without any prefix or suffix cues.
- Co-reference resolution for names will be non-trivial since they have multiple spellings, only context can be used to resolve them. For example, *Milosevic* is spelled at least with three different spellings.
- Honorific titles are very important but a title like *Sadr (President)* can occasionally lead to incorrect recognition because *Sadr* is the location of a well known neighborhood of *Karachi (largest city in Pakistan).*
- Honorific titles are sometimes transliterated into Urdu from English and other times they are scribed in indigenous from in another article to refer to the same person e.g. کیپٹن is the transliteration of *captain* and کپتانmeans *captain* in indigenous Urdu form.
- Anchoring around the named entities is a useful heuristic. The anchor text choice is one of the most challenging tasks for our system.

7.2 Algorithm for Urdu NER

In our rule-based system, the rules form a finite state automata (FSA) based on lexical cues. Some cues are at the start of the state, some are at the end of the state, sometimes the cues are found in the middle of the finite state machine. These rules are corpus-based, heuristic-based, and grammar-based. The rules are implicitly weighted in the order they are applied. For example, the most probable match is listed first and applied first on the text string. For example, one rule that has high chance of recognizing the person name is راہنما - - - - - نے. Here راہنما and نے are anchors. In English this rule will be represented as *Rahnuma* $[token_1, token_2]$ *post-position (Rahnuma* means leader in Urdu). In the example given $token_1, token_2$ will be tagged as a named entity. Each rule is represented as regular expressions since they are an ideal way to represent rules created as finite state automata. Instead of finding the named entities in each document in this version of the system the algorithm finds named entities in the given string of text regardless of the document. The input to the algorithm is a UTF-8 or UTF-16 Urdu text string. One document contains two input strings, the title of the document and the paragraph represented as long string without line breaks. There could be a number of named entities in the paragraph but our rules currently address on named-entity recognition per rule. An n-gram approach was used to limit the length of the input text. After a number of experiments, a 6-gram model was used for an input string. The bigram model was too small and trigram models showed it had no room for named-entities for multi word named entities, four gram and five gram models lacked adequate room for anchor texts and cues. 6-gram model was quite successful but sometimes the windows size was too big when a tri-gram would have worked e.g. two anchor tokens and a named-entity representing one token. The n-grams were constructed by using JDOM implementation to read in XML documents.

When a named entity is found with full confidence it is propagated to all 6-grams and if the matched named entity is found in other input strings (6-grams) it is tagged as a named-entity. Once the 6-gram is tagged it is not processed again. There are some named entities that are abundant in the text but sometimes their occurrences are ambiguous in a number of ways. The reason for these ambiguities is because these entities are so prevalent in the news articles and common in the South Asia that reporters and writers of news articles do not use cues to refer them in the news text. For example, cities like *Karachi, Lahore,* and countries like *America, Bharat, Pakistan* occur frequently with no cues. Instead of writing complicated regular expressions, a small authority file is created with these important names. This authority file serves like a mini gazetteer for our system. A lookup is done before the rules are applied if the name is found the entity is marked. Currently, the authority file contains 40 named entities after examining the 200 document rule-creation set. In the absence of the authority file, complicated rules will need to be crafted using morphological analysis for words like *Pakistan* and through some co-reference resolution for words like *Karachi*.

The complete algorithm is given below:

```
• Iterate over the input 6-grams
  a.  Given the input text match the
    string's tokens with the tokens in
    the authority file.
   b. If the match occurs mark the
      named entity and iterate all
      other input strings and mark them
      with the matched entity if it is
      present.
     i. The strings that are tagged are
        removed from the pool to be
        matched
   c. If the match does not occur in
      the authority file iterate over
      the regular expressions to match
      the expression on the input
      string.
     i.  If the match occurs on a
        regular expression, mark the
        name entity and iterate over
        all other input strings and
        mark them with the matched
        entity if it is present.
```

It is important to note that the algorithm presented above recognizes name entities in the exponential complexity for clarity but the actual implementation is done in linear time complexity.

In the algorithm, regular expressions that are the bottom of the list will be applied when the input string was not tagged with any previous regular expression and the input string did not have any token that is the authority file. The regular expressions that are towards the bottom of the list tend to have patterns that are mostly recognized by the readers who have background knowledge about the topic discussed in the document e.g. the string of names of cricket players without any reference to the *cricket* or *athlete*. English translation of the text would be *Rashid Latif and Shahid Afridi are in the field.* These names of Pakistani cricketers will be known to most South Asians who have followed cricket at any level.

Given an input 6-gram, there could be more than one entity in the input string but we are only finding one named entity and then not processing the string again. This might give the impression that other named entities will not be tagged. Our set up of n-grams prevents us from the missing the later named entity in the string because these entities will show up as one of subsequent 6-grams.

The rules at the top of the list could tie for importance e.g. The rules for جنرل فیصل *(General Faisal)* andشاهراه فیصل *(Shahrah-e-Faisal or Faisal Boulevard)* have very consistent previous token cues. Our strategy of looping through all the 6-grams to tag the named entities is going to tag both strings as named entities but it will not classify شاهراه فیصل as the location if the *"general"* rule was applied first. This has the side-effect of low recall for nested-entities.

7.3 Evaluation & Results

The rule sets were created from 200 documents of Becker-Riaz corpus and the experiment were run on 2,262 documents. Each of these documents is evaluated to create relevance judgments. The relevance judgments are created by two native speakers of Urdu who are avid news readers. The results of experiment runs were hard to grade on such a large set of documents so we chose 600 documents for evaluation. Two judges were chosen who are fluent in Urdu but required some coaching to recognize the named entities. At first judges were expecting terms like *Palestinian* and *elections* to be named entities but after some coaching all evaluation was done correctly. There were very few disagreements among the judges after coaching. A third native speaker was used to address instances of disagreements between the two initial judges. The evaluation set was chosen where all the judges agreed upon the named entities. The results are measured by $f-measure$ that is defined in terms of well known Information Retrieval measures of precision $P$and recall$R$. $f-measure$ is defined by the following equation: $f-measure=\frac{2PR}{P+R}$

Since our algorithm does not support named entity recognition at a document level, the total number of unique named entities in the evaluation set are found. The total numbers of unique named entities are 206. The algorithm matched about 2819 total named entities. While creating the rules and the evaluation set it looked as the number of documents grows the unique named-entities will level out gradually, but we found a lot of repetitions as the number of documents increased but new names consistently were added to the unique list but at a very low rate. Although, the corpus domain is news text, the genre of the documents spans over almost any news worthy information in South Asia, this results in increase of non-unique names. The algorithm execution resulted in 187 named-entities and 171 of those were true named entities. The results show the recall of 90.7% and precision of 91.5%. This gives the $f_1-measure$ value of 91.1%. We found that, suffixes cues and anchor text features were very useful feature but at the same time anchor text feature was the cause of most false positives. Almost all false positives were noun phrases. We ran our rule set on the 36,000 token Urdu data provided for IJCNLP 2008 NER Workshop. Without tuning any of the rules $f_1-measure$ was 72.4% and after adding a few rules after looking at the training set $f_1-measure$ was increased to 81.6% on the test set. A close analysis of this data showed considerable lack of named entities in contrast to the Becker-Riaz corpus. Therefore major results are drawn from the Becker-Riaz corpus. The results of rule execution on IJCNLP 2008 data for Urdu are better than any of the results reported in IJCNLP 2008 NER workshop for Urdu data.

7.3.1 Discussion

Although our results are very encouraging some discussion is warranted about the experience in creating and refining the rules for named entity recognition.

- The 6-gram is processed a number of times to see the performance with stemming and noise words. Both stemming and removal of stop words lowers the precision of the system.
- We mostly used Urdu postpositions as suffix anchor texts. This rule sometimes gave a high recall but very low precision e.g. the postposition conflicted with the transcribed English words in Urdu.
- We removed a rule where the entity is preceded by the punctuation mark colon in the title filed. This rule gave 100% recall but the precision was about 30%.

- Some of the cue words gave 100% recall but the precision was quite low e.g. the rule that identifies name entity through the cue word of transcribed English word of leader gave perfect recall but 56% precision.
- The phrases that could contain more than one token are sometimes written with the blank space between tokens and sometimes as one token e.g. وزیراعظم *(prime minister)*. In this case the rules are modified to recognize both occurrences.

8. Conclusion and Future work.

NER in Urdu is a challenging problem for language processing. In the absence of a learning training set, rule-based approach for NER in Urdu shows promising results. Also, we argue that Hindi resources like gazetteers etc. cannot be used Urdu NER models. Our results are an improvement on all other approaches that are used for Urdu NER. It also shows that our rule-based approach is superior to Conditional Random Fields approach used in IJCNLP 2008 NER workshop by the majority of the papers. In future we plan to use online dictionaries from CRULP through Web Services framework, if available instead of the manually created authority file. Finally, we want to change our regular expressions to accommodate already named entity tagged texts and also to identify names at document level.

9. References

D. Bikel, S. Miller, R. Schwartz, R. Weischedel *Nymble: A High Performance Learning Name-Finder,* Proceedings of 5th Conference on Applied Natural Language Processing. 1996

D. Becker, B. Bennett, E. Davis, D. Panton, and K. Riaz. *Named Entity Recognition in Urdu: A Progress Report.* Proceedings of the 2002 International Conference on Internet Computing. June 2002.

D. Becker, K. Riaz. *A Study in Urdu Corpus Construction.* Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization at the 19th International Conference on Computational Linguistics. August 2002.

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

CRULP http://www.crulp.org/software/langproc/POS_tagger.htm (February 2010)

D. Palmer, D. Day, *A Statistical Profile of the Named Entity Task,* Proceedings of 5th Conference on Applied Natural Language Processing. 1996

D. Matthews, *Urdu in India,* Annual of Urdu Studies vol. 17 (2002).

J. Erickson, *Jack Hermansen on Mulit-culatural Name Recognition Software*, Dr. Dobbb’s Journal July 2005.

J. Lafferty, A. McCallum, F. Pereira, *Conditional random fields: probabilistic models for segmenting and labeling sequence data*, International Conference on Machine Learning, 2001.

L. Wei ; A. McCallum, *Rapid development of Hindi named entity recognition using conditional random fields and feature induction*, ACM Transactions on Asian Language Information Processing (TALIP), volume 2, issue 3. 2003

Mukund, S., Srihari, R., *NE Tagging for Urdu based on Bootstrap POS Learning,* Third International Workshop on Information Access, Addressing the Information Need of Multilingual Societies (CLIAWS3), 2009

Ijaz, M., Hussain, S., *Corpus Based Lexicon Development,* in the Proccedings of Conference on Language Technology. 2007

Goyal, *Named Entity Recognition for South Asian Languages,* Workshop on NER for South and South East Asian Languages, IJCNLP 2008

Ekbal, S. Bandyopadhyay, *Bengali Named Entity Recognition Using Support Vector Machine,* Workshop on NER for South and South East Asian Languages, IJCNLP 2008

P. Srikanth, K. Murthy, *Named Entity Recognition for Telugu,* Workshop on NER for South and South East Asian Languages, IJCNLP 2008

Ekbal, R. Haque, A. Das, V. Poka, S. Bandyopadhyay, *Language Independent Named Entity Recognition in Indian Languages*, Workshop on NER for South and South East Asian Languages, IJCNLP 2008

K. Gali, H. Surana, A. Vaidya, P. Shishtla, D. Sharma, *Aggregating Machine Learning and Rule Based Heuristics for Named Entity Recognition,* Workshop on NER for South and South East Asian Languages, IJCNLP 2008

S. Kumar, S. Chatterji, S. Dandapat, S. Sarkar, *A Hybrid Named Entity Recognition System for South and South East Asian Languages,* Workshop on NER for South and South East Asian Languages, IJCNLP 2008

P. Shishtla, K. Gali, P. Pingali, V. Varma, *Experiments in Telugu NER: A Conditional Random Field Approach,* Workshop on NER for South and South East Asian Languages, IJCNLP 2008

Nayan, B. Ravi, K. Rao, P. Singh, S. Sanyal, R. Sanyal, *Named Entity Recognition for Indian Languages*, Workshop on NER for South and South East Asian Languages, IJCNLP 2008

P. Praveen, R. Kiran, *Hybrid Named Entity Recognition System for South and South East Asian Languages*, Workshop on NER for South and South East Asian Languages, IJCNLP 2008

Chaudhuri, S. Bhattacharya, *An Experiment on Automatic Detection of Named Entities in Bangla*, Workshop on NER for South and South East Asian Languages, IJCNLP 2008

P. Shishtla, P. Pingali, V. Varma, *A Character n-gram Based Approach for Improved Recall in Indian Language*, NER Workshop on NER for South and South East Asian Languages, IJCNLP 2008

R. Vijayakrishna, L. Sobha, *Domain Focused Named Entity Recognizer for Tamil Using Conditional Random Fields*, Workshop on NER for South and South East Asian Languages, IJCNLP 2008

P. Thompson, C. Dozier, *Name Searching and Information Retrieval*, Proceedings of Second Conference on Empirical Methods in Natural Language Processing. 1996

N. Wacholder, Y. Ravin, M. Choi, *Disambiguation of Proper Names in Text,* Proceedings of 5th Conference on Applied Natural Language Processing. 1997.

S. Martynuk, *Statistical Approach to the Debate on Hindi and Urdu*, Annual of Urdu Studies vol. 18 (2003)

R. Russell, *Some Notes on Hindi and Urdu*, Annual of Urdu Studies vol. 11 (1996).

K. Riaz, *Concept Search in Urdu*, Proceedings of the 2^{nd} PhD workshop on Information and Knowledge Management, 2008

K. Riaz, *Urdu is not Hindi for Information Access*, Workshop on Multilingual Information Access, SIGIR 2009

S. Jha, D. Narayan, P. Pande, P. Bhattacharyya, *A WordNet for Hindi,* International Workshop on Lexical Resources in Natural Language Processing, Hyderabad, India, January 2001

H. Wallah, *Conditional Random Fields: An Introduction*, University of Pennsylvania CIS Technical Report MS-CIS-04-21, 2004.