Improving Search via Named Entity Recognition in Morphologically Rich Languages – A Case Study in Urdu

A DISSERTATION SUBMITTED TO THE FACULTY OF THE UNIVERSITY OF MINNESOTA

BY

Kashif H. Riaz

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Adviser: Dr. Vipin Kumar

Co-Adviser: Dr. Blake Howald

Co-Adviser: Dr. Jeanette Gundel

February 2018

Dedication

To my teachers, parents, and Komayl

Acknowledgments

To my late grandfather Intizar Hussain, I will always cherish the time I spent scribing for him. The access he provided to his library was no less than the Book of Kells for me.

I am thankful to my committee members and my advisers for their support on this journey. This work would not have been possible without your encouragement to pursue multidisciplinary research. I thank Dr. Vipin Kumar for his immense support, guidance, encouragement, and most of all that he was gracious despite my procrastination and supported me through the ups and downs of life on this journey. I immensely thank Dr. Blake Howald for his guidance and showing me the path through the fog of this multifaceted research. Most importantly, I am grateful for the generosity of his time, his encouraging comments, his feedback, and taking me through the finish line. My gratitude to Dr. Michael Steinbach for his gentle guidance, his feedback on this dissertation, and components of my research on this journey. He has always listened patiently and talked me through the challenges of my research. Thanks to Claudia Neuhauser, for teaching me how to look into complex societal problems in a multidisciplinary way and giving me hope that there is light at the end of the tunnel for the problems close to my heart. I thank Dr. Jeanette Gundel for her time, advice, and guidance through this journey. For her patience, as she explained linguistic concepts as we sat in her office going over my research.

To the Information Retrieval, and Computational Linguistics research community who guided me in conferences and workshops by providing feedback during my presentations. Specifically, to the presenters at 6th ESSIR (European Summer School in Information Retrieval) in Glasgow in setting a direction to my research. Your guidance to research in Information Retrieval for resource scarce languages was invaluable. My special thanks for Stephen Robertson, and Leif Azzopardi who on multiple occasions encouraged and provided direction. To Karim Darwish, who explained in detail the complexity of writing a stemmer for Arabic. I am truly standing on the shoulders of giants.

To my supervisors and managers who supported me through this journey. Without their support this journey could not have been possible.

Most importantly, I am thankful to my family, who had to suffer through “wasted” snow days, missing fall colors, shortened spring breaks, and at times a messy house, as I was busy in research. We will make up the time!

Contents

TABLE OF FIGURES

List of Tables

1 Synopsis

Search is a task of finding and returning relevant text snippets or documents based on an input query. State of the art search engines, such as Google and Yahoo, are keyword-based. The keyword-based search engines retrieve, score and, for the most part, rank documents based upon the presence of the query term in the user's query. However, the results are sub-optimal (i.e. not retrieving what it is that you would like to retrieve) in two ways. First, keyword search often retrieves a non-relevant document that contains a keyword. For example, when a query contains the query term *plane* documents containing airplane and co-ordinate planes are retrieved, this phenomenon is called polysemy. Second, keyword search misses documents that are relevant but don't contain the query term. For example, a query term of *car* will retrieve documents about cars but not automobiles. The information need was both cars and automobiles. This phenomenon is called synonymy.

One of the challenges of traditional Information Retrieval (IR)[1] is that the terms used in the user query are not the same as they are in the document explaining the same *concept* (e.g. automobiles and other things canonically associated with *cars*). It is observed that less than 20% of the time people use the same keyword for a well-defined object (Deerwester et al. 1990). There are other approaches discussed in chapter 3 that attempt to address synonymy and polysemy and sometimes referred to as Concept Search in the literature (Berry et. al 1994), (Deerwester et al. 1990), (Landauer et al. 1998), (Haav et al. 2002). The state of the art of Concept Search however does not show promising results i.e. it does not find all the relevant documents from the document collection (low recall) and the retrieved documents do not satisfy user issued query (low precision). This problem is exacerbated when a language is more morphologically complex and has many terms that exhibit polysemy and homonymy either through morphology, or through variation in the script of the language. For example, newswire documents about *Osama*

[1] Information Retrieval (IR) and Search are used interchangeably in this document

Bin Laden and *Bin Laden Construction* are retrieved when queried about *Bin Laden* in English. The example shows mismatch of a company name and a person name. This research confirms other works that transliteration of proper nouns from one language to other is one of the major culprits of vocabulary mismatch (Lisbach 2015) (Round 2017). Newspaper articles transliterate *Al-Qaeda* as *Al-Qaida, Al-Qaidah, Al-Qaydah, AlQaeda* and thus they have low recall. The term means a *base* in Arabic and also an *introductory book* in Urdu usage. An Urdu document collection could contain all three uses of the term. Languages that exhibit this level of variability, such as Urdu, are known as Morphologically Rich Languages (Tsarfaty 2010) and will be a central focus of this dissertation.

1.1 Morphological Rich Languages

There is inherently a gap in the performance of text processing algorithms among different languages. For example, using a dependency parser trained on English Penn treebank on a German corpus (Tsarfaty 2010). Observing these gaps through cross linguistics lens at macro level suggests that inherent richness in the morphology of some languages is argued to the cause of performance degradation (Tsarfaty 2010). A contributing factor to this degradation is the impact of having small corpora, which can lack variation, and other issues such as the lack of associated tools such as treebanks or online dictionaries, etc. However, this factor does not fully explain the systematic degradation of performance. The linguistic properties of these languages are significant numbers of word forms, free word order, and use of morphological information to inform the syntactical relations. This dissertation uses the term of Morphologically Rich Languages (MRL) to describe such languages. MRL are difficult to process with out of the box tools, models, and algorithms that are developed for English, e.g., dependency parsers based trained on Penn Treebank for English did to work on Arabic (Tsafarty et al. 2013). Therefore, an Arabic Treebank is needed to develop parsers for Arabic. In general, the tools developed for one MRL are not suitable to process other languages or other MRL.

MRLs are where substantial grammatical information is present at the word level (Tsarfaty 2010). Also, the lexical information for each word form in an MRL may be augmented with information concerning the grammatical function of the word in the sentence, its grammatical relations to other words, cases, inflectional affixes, and so on. Morphologically showing functional information allows for a high degree of word-order variation because strong syntactic constraints are not needed. Furthermore, since lexical items could appear in different syntactic contexts or positions, they could take various forms. This changing of lexical forms results in high degree of word-form variation (Tsarfaty et al. 2013).

While MRL is used in the field of Computational Linguistics and Natural Language Processing it does not necessarily suggest associations that linguists would observe about a given language's morphology. In linguistics, morphology (and associated classification typologies based on morphology) competes with syntax (Bresnan, 2001) and therefore, a rich morphology is associated with non-configurational syntactical phenomena. At high-level, languages that have free word order, use case markers, show discontinuous constituents and exhibit zero-anaphora properties are considered morphologically rich. English is not an MRL, German, is mildly MRL and starts to show situations where out of the box techniques begin to fail. Semitic languages like Arabic and Hebrew are an extreme case of morphological richness (Tsarfaty 2010). The boundaries around the morphological categories of a language are fuzzy, for example, Arabic has a root-pattern schema, and Urdu is considered an agglutinative language. Urdu chooses to inflect a word based on fusional or agglutinative properties based on the context and the influencing language. Urdu is a curious case of an MRL, where an Indo-Aryan language, written in Arabic Script, that borrows words from Persian, Arabic, Sanskrit, English, Turkish, Portuguese and other languages and inherits the morphological properties of the respective languages.

Consider an example of word order and case markers, languages that don't mark cases have rigid word order like English. In English, it will be hard to distinguish between the meaning of a sentence *the cat ate the mouse* from *the mouse ate the cat* if the word order

is not enforced. Urdu has noun case markers; marking is put on a cat to categorize it as a subject of the sentence.

Tsarfaty et al. (2013) discuss challenges of automatic parsing of text in MRLs. This dissertation shows that these challenges extend to Named Entity Recognition and Search domains. Word-form variations to represent the same context introduce polysemy and synonymy during search and name resolution. Modest corpus sizes do not capture adequate word-forms variations to build machine learning models to predict unseen classes. Furthermore, free word order and polysemy between names and other parts of speech make it difficult to recognize names and resolution of concepts to latent structures.

There is explosive growth of data in for languages that we can considered MRLs on the internet. There are numerous international news agencies that buy local stories in their native form and prepare it for consumers through translation. These news articles are usually **about** *who*, *what* and *where*. Some agencies like BBC offer stories in their native language and native script on a language specific Web site[2] with search functionality. Consider English and Urdu examples below which are taken from the UK media giant BBC's website. These examples show *who* and *where* in a bold font. The English sentence has a mention of *organizations* and *location.* The Urdu sentence is about the Catalonia's election, the leader of Catalonia Carles Puigdemont, and his stance on Madrid's rule. Bolded words in Urdu sentence show proper nouns.

> ***Manchester City*** *moved five points clear at the top of the* ***Premier League*** *as second-placed* ***Manchester United's*** *unbeaten run surprisingly ended at* ***Huddersfield Town***
>
> **کاتالونیہ** کہ رہنما **کارلیس پوگیمونٹ** نے کہا ہے **کہ کاتالونیہ میڈرڈ** کی جانب سے خطے پر براہِ راست حکمرانی کے منصوبوں کو قبول نہیں کرے

[2] www.bbc.com/urdu for Urdu and www.bbc.com/persian for Persian

In order to find information about an event inquiry often starts with proper nouns (Shalaan 2014). Searches on Urdu and Persian BBC sites show that results bring back many non-relevant documents, and a search on stop words bring back all documents that have the stop word. In fact, both Persian, and Urdu web sites of BBC have a disclaimer that search results are not reliable. Such a disclaimer is not present for English BBC which suggests that the search capability of at least Urdu and Persian languages on BBC's Web site is not commercial grade.

1.2 Proposition

We propose that names play an important role in determining the "**about**-ness" – a user satisfaction driven concept – of a document (Belew, 2003) Studies have shown that 71% of the queries in search engines contain named entities (Shalaan 2014). Analysis of 500 million query logs of a commercial search engine as part of this research showed more than 50% of the user queries were about names. In the information seeking disciplines, the task of finding names in raw text is called Named Entity Recognition (NER). I anticipate NER as a subtask for Information Extraction (IE)[3] and indexing those entities in an IR engine will improve its performance along with enhancing user satisfaction with the system. Moreover, I plan to show that indexing these named entities natively in morphological rich languages like Arabic or Urdu will improve from the base line considerably. Consider that a query term is Arabic news text with a word الجزيرة *(Al-Jazeera)* can be recognized as organization name or a noun referring island. The correct classification of the term will facilitate extracting the relevant document.

In most situations a name is not only personal name like John, but also refers to an organization (e.g. Walmart), a geographical area (e.g. Minneapolis), or a law (e.g. Patriot Act). For the purpose of this study the scope of names is limited to entities proposed by Palmer and Day (1996), i.e. times, numbers, personal names, organizations, and geographical areas which represent the bulk of named entities with current research.

[3] IE is similar to IR but information seeking performed is without a user query. i.e. no retrieval of documents

These categories are collectively referred to as proper nouns or named-entities. Although named-entities are usually not associated with search engines, a large number of query terms contain proper nouns and many search misses are related to the queried named-entities since the information is **about** entities.

This research shows that if a named entity like *University of Minnesota* is indexed in the searching engine as an organization then non-relevant documents about the state of Minnesota are either not retrieved or ranked lower than the query that contains the term *University of Minnesota.*

Urdu is a MRL and is spoken by a large population of the world. As I embarked on IR research on Urdu, I realized that Urdu is a resource-scarce language with no enabling technologies like corpora, stemmers, stop words and base-line etc. (Riaz 2007). While creating such resources I realized that names were playing an important role in determining the "**about**-ness" – a user satisfaction driven concept – of a document (Belew 2000). Nonetheless, named entity recognition in Urdu is a nontrivial because of its morphology and orthography. To date, concept search does not explicitly address the use of NER to improve its performance. Furthermore, there is no known research reported for Urdu IR discipline besides my peer-reviewed work. The goal of this dissertation is to examine to a degree of which NER improves search, using Urdu as a test case, and applying the results to other Morphological Rich Languages and English.

1.3 Named Entity Recognition (NER)

The atomic element of information extraction — or of language as a whole — could be considered the 'who', 'where' and 'how much' in a sentence. A name matching system performs this task at the surface by performing lightweight parsing and then delimiting sequences of tokens which answer the just mentioned questions (Bikel et al. 1996). Named Entity Recognition (NER) is an example of IE which has become an important step in many other IR tasks. Officially NER task began as a part of the MUC-6 (Message Understanding Conference) whose objective was to standardize the evaluation of IE tasks, but the significance of NER was realized much before MUC-6. Besides MUC, the Text

Analysis Conference (TAC) organized by the National Institute of Standards (NIST) supports research in a number of evaluation tasks that deal with NLP and NER as a sub-task of NLP. Establishment and experimentation in MUC NER shows the importance of NER in IR and IE research.

The goal of NER task is to automatically identify the boundaries of a variety of phrases in raw text and then to categorize the texts identified. There are three categories: 1) TIMEX to recognize time; 2) NUMEX to recognize numbers; and 3) ENAMEX to recognize proper nouns. ENAMEX can be further categorized based on a domain. The NER task tries to categorize ENAMEX entities into geographic location, company names, and people names to name a few.

Generally, names can have innumerable structure in English and other languages combined, names can overlap with other names and other words, simple clues like capitalization can be misleading for English and are mostly absent in nonwestern languages like Urdu. These problems are sometimes mentioned as Structural Ambiguity. Semantic Ambiguity is another type of ambiguity where the metonymy in language can cause ambiguity. For example, State of Maine, where *state* could mean the provincial location or the *situation* of one of the U.S states.

NER is primarily addressed by using two broad techniques: rule-based methods and statistical learning techniques.

- *Rule-based* methods work by creating a set of rules for recognizing named-entities after examining text. These are thought to be quite laborious and expensive to maintain. This is the preferred method where the training examples for statistical methods are not available. These models are usually domain specific and are very reliable with good performance.
- *Statistical learning* is based on a set of examples that are provided to the NER system. Statistical learning models are further classified into generative models, and discriminative models which learn the entity matching rules when sequences of tokens are given as examples. Hidden Markov Model (HMM) is an example of a generative model. Condition Random Fields (CRF) is a discriminative model which is considered state

of art in sequence matching. Both HMM and CRF are graphical models that represent a statistical distribution – more specifically, conditional dependence, over a number of random variables. CRF results for resource lacking languages are below par in the absence of gazetteer and authority files as shown in IJCNLP 2008 NER task on Indian languages (Riaz 2010). Recently, there is a trend to use Deep Learning methodologies which use deep neural networks to solve classification problems. These methods require massive amount of good quality annotated data to learn predictive features from the annotated training set. There are not enough studies to see their viability to recognize named entities in morphologically rich languages.

1.4 Search

1.4.1 Keyword Search

Besides the advances in search engine technology, most searching today is keyword based. For example, Google is the state of the art for search engine technology today. If one searches for the word *car* on Google, there will be no documents found with the word *automobile*[4]unless a document has both the terms cars and automobiles. Keyword search is largely divided into four categories. The following paragraphs briefly explain each approach with some examples.

- *Boolean Approach* is based on logical operators. It is the primary source for information seeking in commercial search engines like Westlaw and Lexis-Nexis. These search engines support complex query syntax and many times it takes years to be fluid in that domain. The following is an example query from WESTLAW: *Cat /s dog & sheep /p wolf & (injur!).* The goal of this query is to retrieve documents that have *cat* within the same sentence as *dog*, and *sheep* in the same paragraph as *wolf*, and finds all the morphological variants of the term *injur*. Besides being popular Boolean-based approaches have a number of drawbacks like large result set, complex queries, and

[4] This search was executed on 8/14/2013 and the first four pages did not show a document that referenced automobiles.

unordered results sets – no ranking etc. This approach is demanding on a novice user but sometimes performs poorly nonetheless even for expert users.

- *Ranked Retrieval* tries to overcome these drawbacks by using statistics on a document collection and query issued by a user. Ranked retrieval engines find documents according to the frequency distribution of the query terms in a collection. If a document contains many occurrences of the query term which is rather rare in the collection as a whole, this suggests that the document might be highly relevant to the query term. Document retrieval is accomplished by computing the similarity between the query vector *q* and the document vector *d*, by using the similarity formula which is like a cosine measure with terms that have weights.

- *Vector Space models* views information retrieval in the light of linear algebra and vector spaces. VSM represents a corpus as a term-document matrix (document-term matrix is also used) in a vector space. The idea is to represent a semantic space in which documents belonging to the same topic remain close to each other. The dimensionality of the vector space is the number of unique terms in the corpus. The documents of the corpus are represented as the vectors in this space. The computation of similarity between the query and the documents is calculated using the standard dot product between the query vector and the document vector. The query is compared with each and every document vector, and the documents above a certain threshold are returned to the user.

1.4.2 Concept-based Search

The Keyword-based approaches do not retrieve document that do not have the query term in it. One of the biggest challenges of IR is that the terms used in the query of a user are not the same as the terms in documents containing the same *concept*. Therefore, the query term may not be indexed by the search engine and it fails to retrieve all the correct documents. The dimensional reduction approach employed by conceptual searching is

mostly achieved through Latent Semantic Analysis[5] (LSA) and its variants (Berry et al. 1994),(Deerwester et al. 1990),(Landauer et al. 1998),(Haav et al. 2002). The heart of LSA is its use of Singular Value Decomposition (SVD). This technique is used for dimensionality reduction of the concept space. LSA can be examined as a special case for knowledge induction, Bayesian regression model (Story 2000), and term co-occurrence analysis (Kontostathis 2003). LSA and its probabilistic variant PLSA is integrated into many commercial products (Deniston, 2003). LSA tries to improve the query mismatch by assuming that the terms represented in a document are unreliable and may be an incorrect representation of the document. These unreliable representations should be replaced by some other set of entities that are reliable representations of the document. Therefore, LSA tries to extract the higher order structures or latent structures in the association of terms and documents to extract such relationships (Deerwester et al. 1990),(Landauer et al. 1998),(Haav et al. 2002).

1.5 Urdu

Urdu is one of the many languages spoken languages in the Indian Subcontinent. It is the national language of Pakistan and is one of the official languages of India – the only Arabic Script official language of India. Urdu speakers range from 100 million to 300 million. Its grammatical rules are different from Arabic – a Semitic language. Therefore, computational models for Arabic are not suited for Urdu. The Urdu alphabet is modified from the Persian alphabet, which itself is derived from the Arabic alphabet. Persian, Arabic and Urdu are considered Morphological Rich Languages. There are alphabet letters in Urdu that have no equivalent representation in Persian, Arabic or Hindi.

In addition to considering computational aspects of the alphabet, Urdu has some unique syntactic properties. Urdu is the only Arabic Script languages of the Indian Subcontinent, but its syntax is identical to Hindi. The primary difference is in the vocabulary and orthography, where Hindi is written in Devanagari script. Urdu's grammar and

[5] Latent Semantic Analysis (LSA) and Latent Semantic Indexing (LSI) are used interchangeably.

morphology is a combination of many languages, Sanskrit, Arabic, Farsi, English and Turkish to name a few.

There are currently no native search engines available for the Urdu language that do stemming, stop word removal or Urdu NER to improve precision and recall. If a document is written in Unicode and available for indexing, it will be picked up by Google or Yahoo by strict Keyword-based match and will be available in the search result list. One of the goals of this research is to build a search engine in Urdu that is named entity aware and improves the baseline.

1.6 Methodology

In this dissertation I will demonstrate that inclusion of named entity recognition on a range of MRL and non-MRL languages improves search. I have focused on named entities that are proper nouns like names, organizations, and locations, dates, numbers. Some MRL languages for research have a rich set of linguistic resources like corpora, stemmers, stop words, gazetteers and some have very few resources. Arabic is a resource rich MRL whereas Russian has some resources like a corpus and a stemmer and no resources were available for Urdu. English is chosen a non-MRL and a standard baseline.

In order to do any experiments in IR or Natural Language Processing (NLP) a number of enabling technologies are required such as corpus, tokenization, stop word detection, stemmers etc. There are currently no native search engines available for Urdu language that do stemming, stop word removal or other interesting language engineering to improve precision and recall. Urdu does not have any of these enabling technologies as do other resource rich languages like English, Arabic, Hindi and Mandarin to name a few. A language that lacks these enabling resources is called low resource or resource scarce languages for automatic language processing. In absence of these enabling technologies, it is necessary to create them for my research. This section briefly describes my contribution in creation of enabling technologies for Urdu. Detailed description of these contributions will be discussed in Section 4.3.

1.6.1 Enabling Technologies

Corpus: Currently, there are only two known Urdu corpora available to the community. One is the EMILLE Lancaster Corpus, in which Urdu is one language among many, and is the more comprehensive of the two (W0038 2003) only in the number of tokens. The other is the Becker-Riaz Urdu Corpus, a corpus of strictly BBC Urdu news articles (Riaz et al. 2002). EMILLE corpus has many typographical errors and has a dearth of named-entities. Becker-Riaz corpus does not have any of those issues. Becker-Riaz corpus is used for NER and NER enhanced search research to explore variations and challenges of named entities in Urdu.

Stop words: A list of stop words is generated from Becker-Riaz corpus and EMILLE corpus, a union of those two lists will be used for stop words. Stop words are considered nonfunctional words (Riaz, 2007)

Stemmer: There are no freely available stemmers available for Urdu that can take a document or a list and stem the words for search engine purposes. A rudimentary stemmer is created that will be used to improve recall. I will show the quality of stemming on the accuracy of search in morphologically rich languages and how it differs than English (Riaz, 2007)

1.6.2 Evaluation Measures

Standard quantitative measures to assess the performance of search engines, concept search and NER systems are recall, precision and F-1 measure. The definition of each is given below:

- Recall is a measure that determines the ability of the search engine to find relevant documents. Technically, it is the ratio of the number of relevant documents retrieved to the number of relevant documents in the collection. It is usually denoted by R. Recall is calculated by the following expression:

$$R = \frac{|relevant \cap retrieved|}{|relevant|}$$

Equation 1

- Precision is a measure that evaluates the ability of search engines to find documents that satisfy the user's need. Technically, it is the ratio of the number of relevant documents retrieved to the number of total documents retrieved. It is usually denoted by P. Precision is calculated by the following expression:

$$P = \frac{|relevant \cap retrieved|}{|retrieved|}$$

Equation 2

- In practice, if a system shows high recall it has low precision and vice versa, this makes recall and precision competing measures. F1-measure is the harmonic mean of recall and precision in that it combines these concepts to give one measure for evaluation. It is usually denoted by F_1 and is expressed by the following expression:

$$F_1 = \frac{2PR}{P + R}$$

Equation 3

These quantities can be explained concretely by the following example in a keyword-based search engine test bed. In this test example documents have the synonymous terms of *cars* and *automobiles*. Let d_1, d_2, d_3, d_4 and d_5 be a collection of five documents and a user executes a query q

$d_1 = \{car, red, fast, insurance\}$

$d_2 = \{automobile, coverage\}$

$d_3 = \{automobile, color, fast, premium\}$

$d_4 = \{cars, classic, liability\}$

$d_5 = \{autobahn, Germany\}$

$q = \{car\}$

Result set of query q is $\{d_1, d_4\}$ when stemming is performed on the words in the collection and only $\{d_1\}$ without stemming. There are 4 relevant documents in the collection.

$R_{stemmed} = \frac{2}{4} = 0.5 = 50\%$ $R_{\not stemmed} = ¼ = 0.25 = 25\%$

$P_{stemmed} = \frac{2}{2} = 1 = 100\%$ $P_{\not stemmed} = \frac{1}{1} = 1 = 100\%$

$F_{1\ stemmed} = \frac{2(0.5)(1)}{1+.5} = 0.66 = 66\%$ $F_{1\not stemmed} = \frac{2(0.25)(1)}{0.25+1} = 0.40 = 40\%$

1.6.3 Base-line for Urdu Search

One of the major hurdles for Urdu processing is the lack of baseline evaluation mechanism for results. Base-lines are TREC like evaluation test bed. There is no Urdu TREC like baseline exists. As a research contribution a TREC like evaluation test bed for Urdu evaluation is created (Riaz 2008). The test bed contains the Corpus of 200 Becker-Riaz documents, information requests or queries, and relevance judgments are created by university students in Pakistan and native news readers in the United States. For this dissertation, 200 documents are used from a much larger set from Becker-Riaz corpus because it is extremely hard to create relevance judgments and topics for thousands of documents with no funding. For Urdu, a base line with various search engines was created (Riaz 2008). The evaluation was done without stemming and stop-words. The results showed that stemming and stop words will be very helpful.

1.6.4 NER for Urdu

The research for NER in the South Asian languages has been quite low mostly because a 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 studies of South Asian languages and specifically on Urdu was done by Riaz et al. (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 (Riaz et al. 2002). 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 NER 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. There

are 15 papers in the final proceedings of NER workshop at IJCNLP 2008. There was not a single paper that focused only on Urdu NER. Within th 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 CRF with limited success. The experiments that showed better results used online dictionaries, gazetteers and other methods to boost performance along with CRF.

Urdu as an MRL has some interesting challenges for NER that are in addition to challenges faced by languages like English. Urdu is written in an Arabic script and has no capitalization and its morphology is complex. It borrows vocabulary heavily from multiple languages tp add words referred to as loan words. Sometimes this borrowing is done with a non-Arabic script language embedded in the script and other times it is transliterated. The text is written with no diacritics and sometimes vowels are omitted. The following example will highlight NER issues with the presence of numerous loan words in Urdu from multiple languages.

Riaz (2007) illustrates this complexity with the composition of the name of Osama Bin Laden which is used in abundance in news media in many languages. بن *(son of)* pronounced as *bin* is an Arabic name cue which needs to be used in the middle of the name for a person name. An Arabic NER system will cue on this pattern and will be mostly successful. Although, بن *(son of)* is an Arabic loan word in Urdu, it and other loan words have homonyms in indigenous Urdu. Following are four senses of the word بن *(bin)* along with examples:

- بن *(son of)* as in Osama Bin Laden a cue of an Arab name that is used in Urdu
- بن pronounced as *bin* means without in Urdu as in تیرے بن *(without you).*
- بن pronounced as *bun* means forest as in سندربن *(Sundar Bun)* – a legendary forest in eastern Bengal
- بن pronounced as *bun* means getting along as in نہیں بن رہی (*It is not getting along)*

- بن pronounced as *bun* means a hamburger bun as in بن کباب *(Bun Kabab)* – a spicy burger

Riaz [96] is the first study in Urdu NER that showed encouraging results – F_1 measure of 92%. Riaz [96] used rule-based approach instead of CRF since there are no other online resources or training sets available for Urdu. Moreover, Shalaan (2014) showed that rule-based systems perform much better for NER on morphological rich languages like Arabic. Urdu named entities are used to enhance the Urdu search system.

1.7 Other Language Evaluation

Besides Urdu, the degree of search improvement NER can have on other languages are evaluated also. This is necessary to understand the general importance of NER on other languages. English, Arabic, Persian, and Russian are evaluated. Evaluation of state of art NER approaches and system in Arabic was performed since it is the most researched MRL Arabic Script language. Persian and Russian languages are morphologically complex and widely spoken but are considered low-resource languages. Since Urdu and Hindi are considered the same language by many researchers, I show that the Hindi resources are not fit currently for doing Urdu research (Riaz 2012). English is chosen as a baseline as it is the most researched in NER and IR discipline. One of the most important criteria for each language evaluation is that a search engine is available to do evaluation in its native script with some enabling technologies (such as stemming and stop words) enabled.

1.8 Experiment Sketch and Results

The ultimate goal of this research is to show that NER can improve Search in general and specifically for Morphologically Rich languages with low resources like Urdu. Before I embarked on research, there were no resources available to do research in Urdu [93], either in Information Retrieval or in general automated language processing. Once these resources were in hand, a base line for each language was created as explained in chapter 4.3. Since one of the search challenges in Search is vocabulary mismatch between query and documents, LSA approach was used to determine if polysemy and synonymy in Urdu

can be addressed through dimensional reduction. The results show that LSA performed poorly compared to standard VSM (Riaz 2008). The lack of standard search systems in Urdu required creating a search engine from ground up and configuring it to be used with and without NER. There is a spectrum of linguistic resources available for each language besides Urdu. For Arabic, Russian and English a search engine is used where an evaluation can be simulated with and without NER enhancement as these engines are not NER-enriched. For each language a set of queries were created with named entities to reflect the **about-**ness of the query. There are no relevance judgments for Arabic, Russian, and English so TREC-like pooling technique is used to create a baseline after creating queries with named entities for each language. There is no capability to re-index the content for Arabic, Russian and English. Hence, effectiveness of NER was tested using a query term boost method that did not require republishing of the corpus, and a fielded restricted method if a field that could be identified as indexed named entity. The evaluation was done by two native speakers. We showed that on average the named entity enhanced search performed better than state of the art search for each of the languages.

1.9 Analysis

Since large number of query terms is **about** names, recognition and disambiguation of names along with stemmer and stop words can improve search quality. This research shows that that NER enhanced search will improve the search quality in each language that was evaluated with an average lift of 18% from baseline when queries contained named entities across all languages.

Research in resource scarce and in a morphological rich language is a challenging task – a number of enabling technologies need to be created which itself is a daunting but necessary task. Urdu NER is more involved than English because of the heavily borrowing done in Urdu, lack of capitalization and the absence of definite articles.

Our research in NER for Arabic, Persian, and Russian showed that statistical learning techniques are highly dependent upon the quality and the coverage of the annotated material and linguistic resources like gazetteers. Stemming can severely impact the performance of NER system for an MRL like Arabic.

This research highlights and provides a roadmap to build a retrieval engine for resource lacking language and more importantly when there is not enough funding available. Also, it shows that the properties of a language need to be kept in mind while building computational models for that language and bag of words approach may not be suitable for morphological rich language.

1.10 Roadmap

Chapter 2 discusses the background, definition, and state of the art of named entity recognition (NER). Chapter 3 discusses the background and overview of Information Retrieval focusing on the **about-**ness, and the challenges of processing named entities correctly for search algorithms. Chapter 4 discusses the proposed methodology, creation of linguistic resources for Urdu to do research in IR and NER. Section 4.2 details a focused review of the Urdu language as it pertains to this dissertation. Section 4.3 discusses the creation and contribution of a number of enabling technologies like Corpus, Stop words and Stemmers and relevant judgements and baseline. Section 4.4 discusses the relationship between Hindi and Urdu in the computational context because of their history together. Section 4.6 discusses the challenges and creation of NER for Urdu. Chapter 5 discusses the NER approaches by other morphological rich languages. Chapter 6 details experiments, results and analysis.

2 Named Entity Recognition

The goal of this research is to understand the utility of Named Entity Recognition in Search. This section describes the named entities for this research, its background, its utility across different applications and as a subtask of information processing disciplines to drive **about**-ness. We will also describe state of art approaches for NER.

2.1 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. Forr example, bankruptcy of the corner shop John & Sons is not as interesting as the bankruptcy of General Motors. This brings us to a very important question. What is a named entity?

2.1.1 Named Entity

In most recent research a name is not only a personal name but also refers to an organization, a geographical area, a law etc. Name matching does not mean matching of all people names, location names and organization names. The proper name identification depends upon the domain, and the application domain. For example, is Patriot Act a name and should it be categorized as a name in the index of a database? In the legal domain Patriot Act is used as a proper name that has an underlying law. Patriot Act should be categorized as a name in legislative documents and otherwise not. For the purpose of this study, the scope of named entities is limited to the names of entities proposed by Palmer and Day (1996), i.e. times, numbers, personal names, organizations, and geographical areas. These categories are collectively referred to as proper nouns. In language processing one is interested in finding *who, where and when* in a sentence (Bikel

et al. 1996). The goal of the named entity recognizer is to find these entities by using different techniques.

2.1.2 The Name Entity Task (Message Understanding Conference)

The noticeable changes in the general world events and the rapid globalization of the industry stress the need for multilingual information extraction (IE). Message Understanding Conference (MUC) was introduced by DARPA as a series of seven conferences from 1990s to 1998. The conference goal was to bring together researchers and practitioners in the field of IE and compete in the development of algorithms. A common evaluation technique and common set of training and test data were used. Named Entity Extraction was first introduced as part of MUC-6 in 1995, and a related conference MET-1 in 1996 introduced named entity recognition in non-English text. The first non-English languages were Spanish, Japanese and Chinese. After MUC-6 named entity recognition became a standard task for most search engines in many forms. After MUC-7 the best performing systems had the *f-measure* of 95% which is very close to human accuracy —*f-measure* of 97%. Currently, expert identification in free text is considered a challenging task pursued by many industries and academia.

The goal of NER task is to automatically identify the boundaries of a variety of phrases in raw text and then to categorize the texts identified. There are three categories: 1) TIMEX to recognize time, 2) NUMEX to recognize numbers, and 3) ENAMEX to recognize proper nouns. The establishment of and experimentation in MUC Named Entity Recognition shows the importance of NER in Information Retrieval and Information Extraction research.

2.1.3 Applications of Name Recognition

- **Media:** A conscious look at media will suggest that proper names play an important role. Nearly every business newspaper and magazine provides a special company name index, and many print names in bold face, this allows readers to spot articles of interest regarding a certain name. While reading news on the Internet, there is almost always a link under the named entity. E-Commerce applications, e.g. stockbroker's

websites always have links to the TICKER symbols and to the companies. These links suggest that name matching, if done automatically, will have many uses. A cursory look at CNN's news Web site shows that proper nouns are found and linked to the authority pages.

- **Information Retrieval:** Search (IR) is the task of finding and returning relevant text snippets or documents based on an input query. Studies have shown that 71% of the queries in search engines contain named entities (Shalaan 2014). Analysis done as part of this research shows half a billion query logs of a commercial search engine exhibit approximately 50% of the user queries were about names. A query term in Arabic news text with a word الجزيرة *(Al-Jazeera)* can be recognized as organization name or a noun referring to an island – Jazeera means an island in Arabic. The correct classification of the term will facilitate extracting the relevant document. There can be many variations of a name that is, Bill Clinton and William Jefferson Clinton identifies the same person. In retrieval systems based on exact match, the user should enter the correct name but this is a serious problem for non-English names. Many systems alleviate the user discomfort by giving them the option to do fuzzy matching by suggesting the possible spellings. Although this approach works well for English and Latin-based languages, it does not work for Arabic Script languages (Lisbach 2015)(Round 2017).
- **Screening:** Documents are indexed for retrieval systems need to handle many name variations. This may be done by cross-referencing all the names in the *authority file* which lists all the possible names using manual effort. Such name matching is mandatory for financial and other regulatory bodies where they will load a list of names from sanctions bodies such as OFAC [6]and match business data against them. On the surface, this looks simple, but the resolution of *ISIS, ISIL, IS,* and *Daesh* to the same organization is non-trivial. Screening is a sub-task of Information Retrieval with similar issues.

[6] The United States Treasury's office of Foreign Assets Control

- **Question Answering (Q&A)** can be considered an advanced form of Information Retrieval (IR). A Q&A system takes a question as input and provides concise answers that are ***about*** the question. NER is used by IBM Watson and other Q&A systems to analyze the questions to identify the relevant answers or constructing passages to provide an answer. The NER system is used to identify the type of the question e.g. identification of a named entity will help in classifying if a question is of type *definition, fact,* or *time*
- **Machine Translation** is a task of automatically translating a text from one language to another. NER plays a crucial role in the conversion of a name into a meaningful representation into the target language. For example, sometimes the input token needs to be transliterated, and sometimes it needs to be translated to the equivalent phrase in the target language (Lisbach 2015). For example, in Urdu, usually people names are phonetically transliterated – *Clinton* to کلنٹن pronounced as /Klintun/ and *United Nations* as اقوام متحدہ pronounced as */aqvam-e-mutaheda/*
- **Text Clustering** is a grouping of documents or terms together in an unsupervised manner. NER can be utilized to improve the grouping based on the ratio of named entities in each cluster. For example, if a TIMEX named entity is recognized in a cluster, then a cluster about events can be grouped together.
- **Navigation systems** play an important role in our lives and their back end systems have large databases that have location names. Reliable creation of these databases is a huge task that cannot be done manually. The connection between points of interests (locations) and removing ambiguity between entities is an essential part of these systems. Recognition of these names in languages other than English is necessary regardless of locations because there are ethnic enclaves in Western English-speaking cities. For example, Devon Street in Chicago has many shops with non-English South Asian names, as does Edgware Road in London with Arabic script or transliterated names.
- **De-duplication.** Lists of people who subscribe to magazines are periodically merged to make a bigger list by telemarketers and other organizations. The merged list often

contains duplicate names. This list usually does not have information like social security numbers, which will make the person unique. Name matching is used to trim the list so extra flyers will not be mailed.

- **Bioinformatics.** In this domain the main interest is to find the name of genes and gene products.

All the examples above illustrate that Named Entity Recognition (NER) can aid in reducing the information need gap and helps a system to find information that is ***about*** users need.

2.2 General Challenges in NER

This section discusses the challenges of automatic Named Entity Recognition. 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 structures in English due to combination from other languages because name can overlap with other names and other words. Furthermore, simple clues like capitalization can be misleading for English and not present in Arabic Script languages like Urdu. These problems are sometimes mentioned as Structural Ambiguity, discussed later along with Semantic Ambiguity. Semantic Ambiguity is much more difficult to resolve than Structural Ambiguity. Names can have variation in spellings. This is sometimes called “Qaddafi problem” because Qaddafi can be spelled in many ways (Wacholder, 1997)(Lisbach, 2015)(Riaz, 2008)

The following table shows some example of names that are challenging for a typical NER system. These examples are focused towards English for motivation but they exist in other languages also. For languages that are Arabic Script and Morphological Rich there are additional set of problems. A detailed set of additional NER problems in Urdu are presented in section 4.5.2.

Description	**Example**

Names overlapping other names	Murphy Oil vs. Murphy Department Store
Names overlapping words	Prime Computers vs. Prime beef
Corporations containing person names	JC. Penny Co. Oracle Corporation
Names containing AND	Atlantis Mill and Lumber Co., Honda and Toyota Motor Company
Ambiguous first names	Bill Clinton vs. Bill text Roth IRA vs. Roth investment

Table 1

Most of the examples mentioned above can be easily identified but it will be very difficult to categorize them accurately as names along with their semantic meaning.

2.2.1 Ambiguity of Proper Names

The goal of NER is first to recognize the potential named entities and then resolve the ambiguity in the recognized name. Like any other natural language processing problem there are two types of ambiguities in names: Structural ambiguity and Semantic ambiguity. Structural ambiguity in English happens because of prepositions (e.g. *Technical Institute of Minnesota*), conjunction (e.g. *Toyota and Honda Motor company),* and possessive nouns (e.g. *Alaska's Sarah Palin*). Structural ambiguity is present in nested entities. For example, *Mahatma Gandhi Road* has a name composed of a person name but represents actually represents a location. Internal structure of the proper name can be ambiguous. For example, consider the sentence "*My teacher in Persian Art Kelley invited me for a conference*". In this case, it is unclear if the teacher's first name is *Art* or *Kelley*.

Semantic ambiguity is another type of ambiguity where the metonymy in language can cause ambiguity. For example, *State of Maine*, where *state* could mean the provincial location or the *situation* of one of the U.S states. Also, *Winona* can refer to singer Winona Judd or to south eastern Minnesota town *Winona.*

Proper nouns resemble definite nouns in what they are referring to can be ambiguous (Wacholder, 1997). For example in an organization Robert Peters can be referred as Bob Peters or sometimes R. Peters. It is difficult to figure out that all variations of an entity refer to the same person. This can be resolved by different levels of effort in solving the co-reference problem. Non-English names pose another dimension of problems in name recognition. For example, the most common first name in the world is Muhammad, which can be transliterated as Mohmmed, Muhammad, Mohammad, Mohamed, Mohd and many other variations. Variations of names and organizations are a major issue in reducing the accuracy of a search engine.

There are some ambiguities which are particularly interesting in names. For example, when referring to cook one can infer a chef or the name of the person Cook or Cook County in Illinois. Capitalization usually solves the problem, but when the word appears in the start of the sentence, resolution become a little tedious. Capitalization does not always solve the problem. For example, New Coke and New Sears suggest the same pattern, but world knowledge can disambiguate those two (Wacholder, 1997).

2.2.2 NER as an Information Extraction task

NER can be introduced as a sub-task of Information Extraction (IE). The goal of IE is to fill a template like structure from the information gleaned from the unstructured text. The template is composed for entities and relationships about entities. The relationships are the events or facts around the entities. Generally, after the template is filled by the IE system no new information needs to be gleaned about the paragraph. In other words the template completely fills the *what, who and where* of the sentence. An example of the *event* is Pakistan's army move into the Swat Valley, and a *fact* is the defeat of BJP in Indian elections in 2009.

Information Extraction can be thought of as a pseudo-document understanding system. In an IE system, there is no attempt made to do discourse analysis or deep parsing. Instead before performing the IE task, the set of parameters is defined regarding the kind of information that is sought by the system. For example a system that is written for trend

analysis will not fare well on named entity recognition system. The simplest form of information extraction is *term* extraction, where the IE module extracts the simple terms from the input documents (Feldman 2007). For the information extraction module doing NER, a template is filled with the MUC template:

a) people names, organizations, geographical locations
b) dates and times
c) monetary amounts

Feldman (2007) defines NER as a weakly domain dependent task. This means that if the domain of the text changes, the system performance may or may not degrade. Almost always the performance of the system is directly related to the effort put in the system for generalization of the domain. The distribution of the named entities for the MUC conferences have been the in the following order: 70 percent proper names, 25 percent dates and time, and 5 percent monetary amounts. Given the proper names the distribution 40-50 organization names, 12-32 are location tags and 23-29 are percent tags. In the MUC there were a set of names that were not considered to be proper names because they are laws that are associated with people names and newspaper names. Although this is suitable for MUC, this kind of restriction will degrade practical systems in a legal or law enforcement domain. For example, *Megan's Law* or *Miranda* cannot be disregarded because it will reduce the practicality and commercial value of the system.

2.2.3 Evaluation Issues in IE and in NER

The main challenges while doing the evaluation of an IE system are discussed below. Some of the points follow the discussion in Feldman (2007). The biggest challenge for the IE evaluation is to understand the settings where the experiments were conducted. Some variables that need to be looked at are:

- If all the participants used the same corpus, if the corpus was annotated, if annotated who annotated it, if the annotators were linguists, etc.

- If the corpus was annotated what kind of machine learning algorithms were used. What was the size of the test set? How big was the training set? Were other machine learning features like folding, cross-validation, or boosting used?
- The judging of the results becomes an interesting exercise when the system only misses extraneous information like punctuation.
- If the system finds the entities in each document consistently or finds it once and then does not address it.
- The use of enabling technologies like stemming and POS used upon the input set. If the stemmers existed already or were crafted for the exercise.

While doing Evaluation of NER following additional issues need to be looked at:

- The existence of resources like corpus and annotated corpora. This becomes important for scarce resource languages like Urdu, Persian, Russian in evaluation.
- The enabling technologies (e.g. stemmers, stop words, morphological analyzer, or POS tagger). For low-resource languages these resources do not exist and can hurt the performance.
- Lack of annotators and judges for resource scarce languages.
- Availability of supporting information like gazetteers

2.2.4 Architecture of the NER System

NER systems are generally composed of the following components:

1. **Tokenization**: Input document is broken into sub-components: like words, sentences, paragraphs. Boundary detection of an MRL is non-trivial and considerable amount of research is done only on this topic (Hassain 2011)(Ijaz 2011)(Syeda 2013)

2. **Morphological / Lexical analysis:** Once the subcomponents of the document are available, there is an opportunity to analyze the words or sentences like parts of speech tagging, stemming, stop word removal, or analysis of a word for suffixes or prefixes or infixes.

3. **Syntactic analysis**: Usually Information extraction systems do not do deep syntactic processing. Instead, most of the interesting information can be found by shallow parsing like bi-grams, tri-gram creation.

4. **Domain analysis:** This task is considered one of the hardest components that may need to do deep analysis by combining the information from different part of the document or across documents, or across sentences to build the relationships.

These components are usually part of any text processing system that tries to glean some sort of meaning out of the unstructured text.

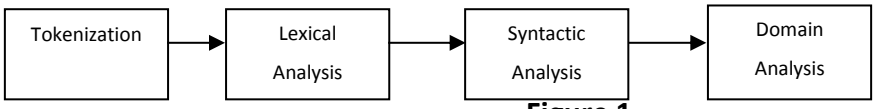


Figure 1

The first two components of the system besides segmenting the document — words, sentences and paragraphs — perform some basic lexical analysis by looking into gazetteers of the census bureau or phone dictionary and other people dictionaries. They match the suffixes like "Co." to signal the end of the organization named-entity.

Named Entity Identification is usually the next step after lexical analysis. Besides using the statistical modeling techniques, Regular Expression patterns are most effective in recognizing the proper names. These patterns include capitalization in English, Honorific words like Mr. and Mrs. etc., and title suffixes like Jr. and Sr.

After the identification of the named-entities, relationships among entities are created through shallow parsing or through deep analysis that may include techniques such as anaphora resolution or co-reference resolution.

2.3 Name Recognition Systems and their Approaches

Over the years many systems have been crafted to find names in different domains. Some of them 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 domains

foreign to them. On the other hand, the systems that claim generality do not work as well as the best domain specific systems but fare poorly when the domain is changed.

Named entity systems either find already known names or follow the pattern when new names can be discovered. The systems that find the existing names can use a brute force approach where the names are looked up into gazetteers or databases. These systems are quite tedious to build and do not provide good predictive information. This approach usually fails in domain specific areas when the coverage is not good. The systems that find new names usually follow a pattern that is inferred from the machine learning process or they follow a set of rules. These systems, given enough examples to train or craft rules, perform very well.

Although a number of name recognition systems have been implemented for English and some show (over time) accuracy in the range of 90-95% (close to human performance), the name-recognition task is far from understood. The systems that show accuracy of 95% start to underperform when some basic feature like punctuation or capitalization from token is withheld. Moreover, if the corpus contains transliterated information, the performance degrades drastically per my evaluation of English in South Asia focused BBC corpus. The systems that use a training set depend on the accuracy of the annotators since the context changes in the domain considerably. For example, the context of Yugoslavia was quite different in 1980 than in 2000 because Yugoslavia does not exist anymore. Named entity can be used as a sub-task of Information Retrieval than as the task of Information Extraction, as was shown by Dozier and Thomson in the legal domain (Thompson, 1996).

What follows is the discussion about the name finding approaches of some of the early NER systems and their modern or commercial incarnation.

2.3.1 Nymble: a High Performance Learning Name Finder

Nymble is one of the systems developed at BBN which claims to have the accuracy of 96% or higher for Named Entity Recognition task. Nymble uses a variant of HMM (Hidden Markov Model) to perform Name Recognition task. Nymble takes the approach that every

document is composed of names and noise, and when it passes through the system all the noise is removed and names can be obtained. Words are considered to be ordered pairs composed of word and word feature. Word feature is simply a category of the word as it is encountered. *Twodigitnum* to represent 90, *fourdigitnum* to represent 1990, *allCaps* to represent BBN, are three of the fourteen categories. Nymble is a purely generative statistical model for named entities. Since its inception, Nymble has become a commercial name recognition engine available under the trade mark name of *IdentiFinder*.

2.3.2 NetOwl™ Extractor from IsoQuest

NetOwl is one of the premier name matching tools that is available commercially and its claim to fame is through entity extraction. Its parent company is SRA Technologies, which has a number of other commercial text mining software tools.

NetOwl uses a set of heuristics that are based on the structure of names like suffix and prefix clues in order to recognize names (e.g. *Inc.* for organization). It uses linguistic knowledge that identifies the context in which names can appear (e.g. corporate executives are often described along with a title, other descriptive information, and the name of their company).

NetOwl Extractor consists of a software engine that applies name recognition rules to text, which are supported by lexical resources and limited lists of proper nouns. A name recognition rule consists of patterns and actions. Patterns are simply regular expressions, and actions categorize the names with tagging and classification. NER system's rules are segmented into phases. For example, one phase recognizes personal names and other organizational names. The lexical resources contain information about words such as parts of speech and their meaning. The following table lists some examples of rules of NetOwl Extractor.

Type of Rule	Pattern	Action	Example
Structural	capitalized personal first name + capitalized word	Tag Match as PERSON	George Bush
Contextual	personal title + capitalized word	Tag match as PERSON Excluding title	Mr. George Bush
structural	capitalized word sequence + corporate Indicator	Tag match as ENTITY Company subtype	Oracle Corp.

Table 2

NetOwl Extractor uses a name list that contains those names which are problematic and can be tagged with a static lookup of names, such as nations, U.S. states, and household acronyms like 3M. NetOwl Extractor claims to have the highest accuracy for recognizing names and linking of similar entities in the industry. NetOwl extractor claims to have NER systems for Arabic, Persian and Cyrillic languages. NetOwl Extractor does not have a NER system for Urdu.

2.3.3 Nominator: IBM T.J. Watson Research Center

Nominator (Wacholder, 1997) is a fully implemented module for proper name recognition developed at T.J. Watson Research Center. Nominator has a good balance of high speed and accuracy. It contains no syntactic contextual information. It applies a set of heuristics to a list of words based on patterns of capitalization, punctuation and location within the sentence. The context information used by nominator is that proper names co-occur in the document. It also recognizes and connects different name variants.

Nominator first identifies proper nouns to build a list of candidate names for a document. It then applies splitting heuristics to break the complex names into smaller names. Next, it groups together name variants which refer to the same entity. After information about the names and their referents has been collected, names are aggregated through a process and are stored in a dictionary. Ambiguity resolution at the start of the sentence is done by keeping all the words at the start of the sentence as candidate names. If the candidate names occur again, but not in the start of the sentence, they are kept as a true name. Otherwise they are discarded. An example of ambiguity resolution is when J. Edgar Hoover and J. Edgar Hoover Building both appear in the text. J. Edgar Hoover is given a high score for a person and lesser score for a place and vice versa. Nominator is no longer available as a separate module. It is folded into DB2 text miner, DB2's text mining component.

2.3.4 Nationality Specific Methods

In comparing names, it may be useful to take into account what one knows about the country or language of origin. For example, if a name comes from a country which typically places a family name before an individual name, language specific rules can be applied to get encouraging results. This can be very useful for looking for Non-English names in the English text. The most prominent effort using this approach is the technique developed by Dr. Jack Hermansen at Linguistic Analysis Systems Inc. where he used the system for State Department, Immigration and Naturalization Services (INS) before it became part of Department of Homeland Security (Erickson, 2005). A detail analysis of

language specific NER shows the MRL have a very complex structure of names with polysemy. Transcription issues may be able to detect if a string is a name but will fail in identity matching (Lisbach, 2015).

Hermansen' s approach consists of two major steps:

1) Classify names according to nationality. Hermansen used four major nationalities or ethnic groups: Chinese, Hispanic, Arabic and Russian.
2) Apply nationality specific rules to the names once that nationality has been determined.

This approach has the advantage of using the knowledge about the phonetics, spelling conventions and other aspect of names. Though useful, this technique may have ethical and legal risks.

2.4 Conditional Random Fields

In general, Conditional Random Fields are considered state of art technology doing NER using statistical learning. Therefore, it is important to analyze what are conditional random fields and how do they work to infer rules from text.

Conditional Random Fields (CRF) was introduced for segmenting and labeling the sequence data as a graphical model (Lafferty 2001). Graphical models are used to represent a statistical distribution over a number of random variables by a product of local functions that depend on small number of variables. Since its introduction, it has been used by a number of Information Extraction and Information Retrieval tasks like Named Entity Extraction and labeling the gene sequences in Bioinformatics (Wallah 2002). CRF is an example of the discriminative statistical model of Machine Learning. It is very closely related with Maximal Entropy and addresses some of the drawbacks of that model. Before the introduction of Conditional Random Fields, generative statistical models like Naïve Bayes and Hidden Markov Models have been used for tagging the sequence data. Within all experiments at IJCNLP 2008 for NER in South Asian languages, Urdu was the most under represented. All algorithms using Condition Random Fields performed poorly on Urdu with *F-measure* in the range of 25-35. However, Hindi and

Bengali consistently measured in the range of 60-80 mostly because of the availability of lookup dictionaries called gazetteers.

2.4.1 Background

Segmenting and labeling a sequence of information is at the heart of many computational sciences such as, Parts of Speech tagging in Natural Language Processing, and sentence or word boundary detection in Chinese language processing. Pre-processing of information in this way can be very useful and augments the main task of understanding the information. For example, Named Entity Recognition can be very useful in increasing the Precision and Recall in Web searches. In order to understand CRF thoroughly, it is important to discuss the models that are used for its inspiration. These are Hidden Markov Models and Maximum Entropy Markov Models.

2.4.2 Hidden Markov Models

In general, processing of sequenced data has been done under the umbrella of Information Extraction (IE) or in Computational Linguistics. For the most part, the most favorable statistical technique used in Information Extraction has been Hidden Markov Models (HMM). Hidden Markov Models are probabilistic finite state machines that represent the generative statistical models of learning from examples. They are based on the Markov process with unknown parameters. They are generative in the sense that the technique finds the best distribution to generate the observed variables (features). In general given a set of observation (examples), this predicts the labels that should be assigned to them. Hidden Markov models define a *joint probability distribution* over random variables $X \; and \; Y$ denoted by $p(X, Y)$ where X are the observations in the sequence, and Y are the labels that may be predicted given the observations. In the generative sense, this finds the label Y that maximizes the joint probability $P(X, Y)$ for the observed variable X. In this model the labels are the *hidden* states. Pictorially, HMM can be represented as the directed graphical model as shown below:

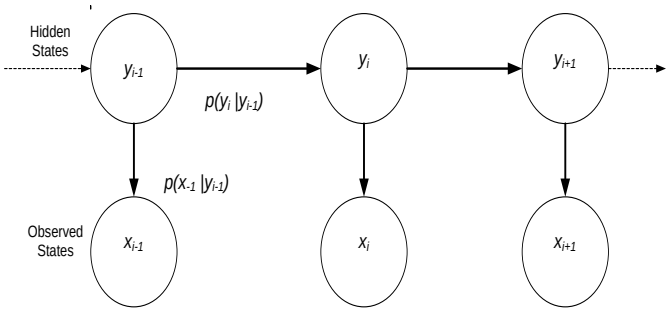


Figure 2

The above figure can also be written as in the following equation:

$$p(x, y) = \prod_{i=1}^{n} p(y_i|y_{i-1}).p(x_i|y_i)$$

Equation 4

As we can see from the description above that HMM looks at only one state at a time

Rabiner (1989) describes three problems that Hidden Markov Models try to solve.

1. Given the observation sequence $O_1, O_2, O_3, \ldots . O_n$ and a model $\lambda = (A, B, \pi)$, how to efficiently compute $P(O|\lambda)$ i.e. the probability of the observation sequence given the model.
2. Given the observation sequence $O_1, O_2, O_3, \ldots . O_n$ and model λ how to choose the transition state sequences $Q = Q_1, Q_2, \ldots . Q_n$ that best explains the observation.
3. How to adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$

In order to calculate the joint probability distribution for this task, the number of possible outcomes to consider may be computationally prohibitive. Therefore, HMM uses independence assumption in the observation variables. This independence assumption is a bit different than the Naïve Bayes assumption; more precisely, in HMM each observed state can only see its associated label state at a time. Although this independence assumption works reasonably well in small and constrained domains, it introduces bias in large complex domains like Bioinformatics and language processing.

2.4.3 Conditional Models

The independence assumption often impedes building a practical model because although sometimes the dependency among variables can shed light on deep structures in the sequence, at the same time a model should be tractable. Conditional models are the statistical discriminative model that alleviates the problems mentioned earlier.

Instead of the joint probability distribution $p(x, y)$ over both the labels and observations, define $p\ (Y \mid x)$ conditional probability over the random variable $Y$of labels given a particular observation sequence x. The conditional models are used to assign a label y' when a new observation sequence x' is encountered. The label y' is chosen such that it maximizes the conditional probability $p(y' \mid x')$. Conditional models are not generative because no attempt is made to model the observations given the labels, and the related characteristics of the observation are captured without explicitly modeling them.

Conditional Random fields are inspired by an earlier model called Maximal Entropy (ME) and Maximal Entropy Markov models and try to address some of their drawbacks. This section briefly discusses Maximum Entropy. Entropy is the measure of randomness; therefore, the measure of Maximum Entropy leads the system towards the maximum randomness under certain constraints. In other words, a model built like that will be most general. Maximum Entropy is used to estimate the probability distribution from the training data. This distribution is maximized when the distribution in question is as uniform as possible.

2.4.4 Maximum Entropy Markov Model (MEMM)

Maximum Entropy Markov Model (MEMM) is a directed graphical model that is very closely related to Hidden Markov models. Like HMM, it is a probabilistic finite state machine. Unlike, HMM which depends upon observations and labeling probabilities, MEMM only depends upon previous label's probability, but that probability is derived from the input observations.

Given observation sequence$X$, MEMM applies the maximum entropy model to the sequence as in assigning label from the random variable Y to each position in the

sequence. The label is generated by considering the probability of the previous label and the observations. The probability of the sequence is the product of the probability of each label state. The conditional probability is represented as follows:

$$p(y|x) = \prod_{i=1}^{n} p(y_i|y_{i-1},x)$$

Equation 5

Pictorially, we can represent the MEMM as follows:

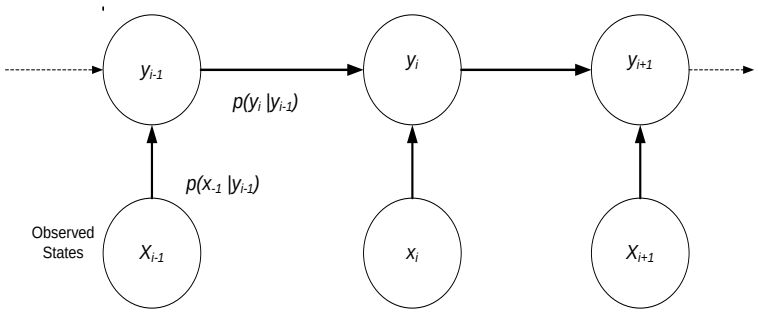


Figure 3

At each position, a local model assigns labels based on observations and previous label. Normalization also occurs at each position and a probability distribution occurs on each label:

$$p(y_i|y_{i-1},x) = \frac{1}{Z(y_{i-1},x)} \exp\left(\sum_k \lambda_k f_k(y_{i-1},y_i,x)\right)$$

Equation 6

λ is the learned weight from the training and f is the functions that applies to the label state, Z is the normalization constant, defined as:

$$Z(y_{i-1},x) = \sum_{y'} \exp\left(\sum_k \lambda_k f_k(y_{i-1},y',x)\right)$$

Equation 7

The local normalization at each label state introduces a regional/local bias in labels with fewer degrees are preferred over those many, and these labels tend to ignore observation. In order to address this local bias, we need to normalize over the whole sequence and not on each local state. Conditional Random Fields try to address the local bias problem.

2.4.5 Conditional Random Fields (CRF)

Conditional Random Fields are conditional probabilistic models that are used for labeling and segmenting sequential data. In contrast to HMM and MEMM, they are undirected graphical model that define one log-linear relationship distribution over the sequences of labels given a particular observation sequence (Wallah 2004). Unlike HMM, CRF do not make any unwarranted independence assumptions, and unlike MEMM CRF, do not make any local biases; normalization is done on the whole sequence rather than each local state. We can represent the graphical model in a graph nomenclature as $G = (V, E)$ where each vertex $i \in V$ represented by the component Y_i of the random variable$Y$. If each random variable $Y_i$obeys the Markov property with respect to G then (Y, X) is a conditional random field conditioned on$X$. We can represent the conditional probability as $p(Y_i|X, Y_j \ \ j \neq i) = p\ (Y_i|X, Y_j, j \sim i), where\ j \sim i$ means that j and i are neighbors in G (Lafferty 2001). Most segmenting, and sequence label applications can be addressed by the most simplistic form of CRF called linear chain CRF where the nodes of Yrepresents the linear first order chain as shown in the figure 4 below. This structure of CFR model generalizes the finite state machines like HMM and MEMM.

The probability distribution $p(Y, X)$ can be represented by the product of the probability of each node represented by Y_i of Y in the graphical model. This is represented by the normalized product of positive, real-valued functions called potential functions. Each potential function works on the subset of random variables represented by the vertices of the graph G. Since product of real-valued positive functions can be greater than 1, normalization is used to ensure the probability axioms are not violated. If an edge is missing between random variables of G say Y_k and Y_l, it means that these random

variables are conditionally independent given all other random variables in the model. This can be accomplished by making sure that the independent random variables do not operate on the same potential function. We do this to ensure that each potential function operates on the set of random variables whose corresponding vertices form the maximal clique within G. In a linear chain CRF, each function operates on a pair of adjacent label nodes Y_i and Y_{i+1}.

Formally, we define CRF to be defined on random variables (X, Y) and a vector of local functions or features $f = (f_1, f_2, \ldots . f_n)$ and a corresponding weight vector $\lambda = (\lambda_1, \lambda_2 \ldots \lambda_m)$. Each local function is a real-valued positive feature of the observation of sequence$X$. The labels of the sequence are denoted by $Y = (y_1, y_2 \ldots . y_n)$ and the specific position at sequence is denoted by i. The value of the feature function at any given sequence position i may only depend on $y_i$or on $y_i$and y_{i+1} and no other label sequence of the random variable Y. The feature function that depends only on y_i at the sequence is called a *state feature function*, and if it depends upon $y_i$and y_{i+1}it is called *transition feature function.* The state feature function of label at position i and the observation sequence x can be represented as $s_k(y_i, x, i)$, and the transition feature function of the entire observation at position i and $i - 1$ can be is in the label sequence is represented as $t_j(y_{i-1}, y_i, x, i)$. It is important to note that we can also represent the transition function for positions i and $i + 1$ as $t_j(y_i, y_{i+1}, x, i)$. We used the i and $i - 1$ convention to model it as HMM and MEMM. We represent $f(x, y, i)$ as feature function that represents both state and transition feature functions. The global feature vector $F(x, y)$ that the represents sum of all the local vectors is represented as:

$$F(x, y) = \sum_{i}^{n} f(x, y, i)$$

Equation 8

Given the system above, the conditional probability for the label y in the sequence given the observation sequence x is written as:

$$p(y|x,\lambda) = \frac{1}{Z(x)} \exp(\lambda.F(x,y))$$

Equation 9

Where $Z(x)$ is a normalization parameter; and λ is the parameter that is learned by training data. The normalize sums over all possible label sequences and is represented as:

$$Z(x) = \sum_{y} \exp(\lambda.F(x,y))$$

Equation 10

The CRF shown above in an exponential form is a consequence of the fundamental theorem about Markov Random fields that state that any conditional probability distribution that exhibits the Markov property i.e. $p(y_i|x, y_j \; where \; j \neq i) = p(y_i|x, y_{i-1}, y_{i+1})$ can be represented in an exponential from with appropriate feature functions and the weight vectors.

Feldman (2007) describes three main uses of CRF like the three uses of HMM describes earlier:

1. Given an observation sequence x and the label sequence y, CRF determines the conditional probability $p(y|x)$.
2. Given an observation sequence$X$ and the label sequence Y, CRF determines the most probable label sequence $y = argmax_y p(y|x)$.
3. Given the training samples in the form (x_k, y_k) find the learning parameter λ that maximizes the likelihood of the training data.

Pictorially, CRF is shown as an undirected graph as follows in two forms.

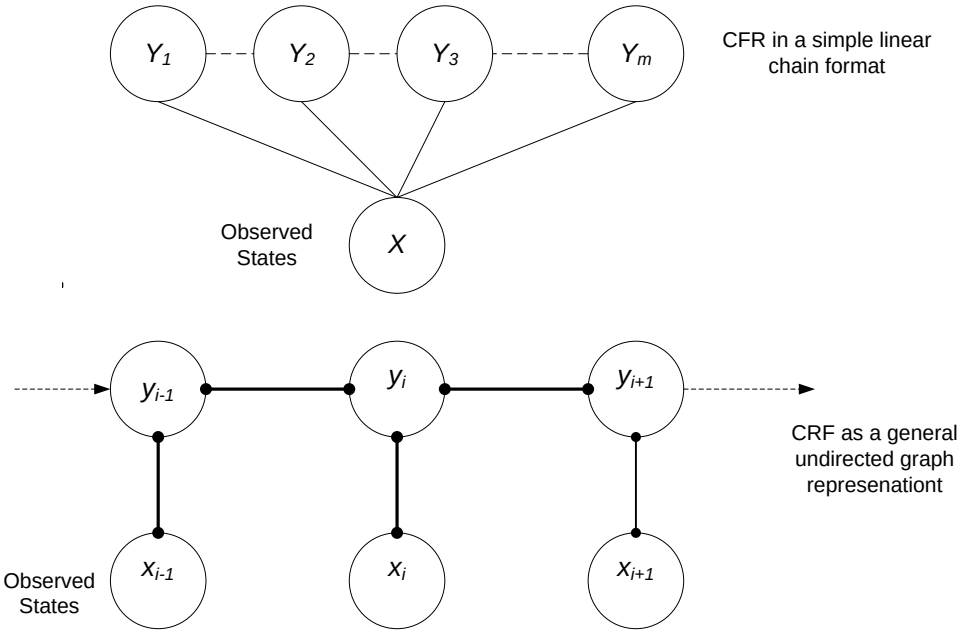


Figure 4

2.4.6 Use of Condition Random Fields for NER

CRF are used extensively for Bioinformatics and for language processing tasks like Parts of Speech tagging and Named Entity Recognition. It was first tried on non-English Hindi data by Li (2003) data since its introduction. The results showed f-measure ranging from 56 to 71 with different boosting methods. In contrast, English and other European languages NER systems have shown results in the range of 90 to 95 using HMM, MEMM and CRF. The main requirement for Condition Random Fields is a good set of training data. Research in Persian, Arabic, Farsi, Hindi and Russian showed that the quality of the training data and quality of features for CRF is of utmost important. For all languages, external resources were used to boost the performance of CRF and the low accuracy was attributed to the lack of quality and coverage of these lookup dictionaries and gazetteers. Among the experiments performed at Named Entity Workshop on various Indic languages and Urdu, almost all experiments used CFR with limited success. The workshop provided training data for all the languages but all better performing systems used their own tagged resources from various Indian institutes. In addition all systems tried to use gazetteers to

boost the performance. In short, none of the systems that used Conditional Random Fields had good results when tried on South Asian languages in contrast to English. Although CRF is the dominant method for NER, in practice CRF runs as good as MEMM as shown in multiple experiments (Shalaan 2014) (Manning 2010).

2.5 Recent Trends in Statistical and Machine Learning Approaches

Besides Markov Model-based approaches, other Machine Learning approaches have been used for Named Entity Recognition (NER), e.g., Support Vector Machines (SVM), Logistic Regression, Reinforcement learning, and Semi-supervised learning. All of these approaches require good quality annotated data.

Recent trends in Natural Language Processing and Machine Learning are focused on Neural Network based approaches, relying on the use of word vectors from large corpora (e.g. word2vec (Mikolov et al. 2013), GloVe (Pennington et al. 2014)) in various architectures (convolutional, recurrent, etc.). While other machine learning applications can work with the modest amounts of data, Deep Learning methods excel (and require) large amounts of data to predict a class based on inputs. In this section, I will briefly describe recent and continuing trends in deep learning and their relationship to the techniques described in this dissertation on the processing of MRLs.

2.5.1 Word2vec

Traditionally, ML techniques treat words in a corpus, document or even a sentence as a bag of words and therefore introduces polysemy and synonymy. There are multiple techniques to remedy this situation which are based on Vector Space Models (VSM) and its derivative Latent Semantic Analysis (LSA). LSA is a dimensionality reduction technique based on VSM that suggest words that appear in the same context share semantic meaning. LSA uses Singular Value Decomposition to create word embeddings in a reduced vector space. The details of VSM, LSA, and vector decomposition are provided in Chapter 3.

Word2vec is a dimensionality reduction technique that tries to achieve the semantic representation in a machine learning based predictive model. Word2vec attempts to

suggest a word through its neighborly associations as it learns weights. These weights are considered the embedded vectors (or word embeddings) or the parameters of the learned model (Mikolov et al. 2013). Word2vec can be used in two ways:

a) Predict a ***word*** given the ***context*** using Continuous Bag of Words (CBOW) technique.

b) Predict the ***context***, given the ***word*** using Skip-gram technique.

Usually, CBOW converges smoothly on a smaller data sets, in contrast to Skip-gram model which works better in larger data sets. CBOW and Skip-grams are trained using a logistic regression function by discriminating the positive examples from the negative examples. The output of word2vec is morphed vector with real-valued numbers representing the words that were sent in as an input. These numerical numbers are called word-embeddings. In all Deep Learning models, the input is transformed into word embeddings before proceeding further in the hidden layers.

2.5.2 Deep Learning

Deep Learning models and architectures are based on Artificial Neural Networks (ANN) that try to simulate how a human brain learns. Usually, an ANN consists of an input layer, one or two hidden layers, and an output layer for classification. In short and at a high level, Deep Learning architectures' hidden layers range from two up to tens or hundreds or more. Each hidden layer adds more sophisticated learning and makes the network hierarchical. The lower layers of the network can discover sub-word features, and the layers above can be used to discuss word related features. The higher layers closest to the output layer can be used to take the input from the lower level nodes to identify named entities. Deep learning architectures are also known as Deep Neural Networks (DNN) indicating multiple levels of hidden layers (LeCun et al. 2015)(Schmidhuber, 2015).

The promise of Deep Learning techniques is to learn the features from the massive amount of data instead of doing feature engineering by hand. Significant studies that reported using deep learning technique continue to have manual input as features.

There are various architectures of Deep Neural Networks each addressing challenges in machine learning problems. The designs that are prevalent in literature are:

a) Convolutional Neural Networks (CNN)

CNN are used to classify images, pattern recognition, and optical character recognition (OCR) (LeCun et al., 2015). They can be used in language analysis tasks to read complex orthography of a language, although no implementation in this capacity has been reported.

b) Long Short-Term Memory (LTSM)

LSTM networks are the most commonly used variation of Recurrent Neural Networks (RNN) in Natural Language Processing applications. In 1990s German researchers Sepp Hochreiter and Juergen Schmidhuber proposed it as a solution to the vanishing gradient problem (Hochreiter et al. 1997)

RNN is in the family of feed-forward neural networks with the property of feedback loop like a human brain to learn from past examples. These type of ANN are historically challenging to train and suffer from a *vanishing gradient problem*, but the use of modern GPU to control the speed of training has somewhat alleviated the problem in addition to the gated architecture. LSTM solves the problem of remembering everything that it has encountered in the past by remembering useful information and forgetting useless information by using a gate-based design that consists of input gates, forget gates, and memory cells. The gating structure allows information to be retained across loops and consequently enables gradients/weights to flow across these loops/layers. This mechanism enables the LTSM model to overcome the vanishing gradient, a known issue with most Recurrent Neural Network models (Schmidhuber, 2015).

2.6 Reflections on Deep Learning and Word Embeddings for Morphologically Rich Languages

Since deep learning technique are so pervasive in recent publications and industry, the following reflections explore if deep learning is as effective on MRL as English.

a) One of the most critical components of Deep Learning is a massive amount of annotated training data set. It remains to be seen the quality of training data on the trained models. As conventional wisdom suggests, better the quality and distribution of training data the better the results and vice versa. Most of the Morphologically Rich Languages for this dissertation do not have NER rich annotated data including Arabic that tends to be resource rich in other tasks.

b) Wordvec or word embeddings, in general, is a dimensionality reduction mechanism to create vector representation and therefore bears similarity to LSA, and vector decomposition approaches. Altszyler et al. (2016) showed that the quality of word2vec requires a massive amount of data to outperform LSA and for smaller corpora LSA has better results to find underlying concepts. Word embeddings are commonly used as an input to a DNN. Therefore, quality of embedding is critical for a performant model. Dimensionality reduction techniques on Urdu, a Morphologically Rich Language showed many false positives because of polysemy and synonymy in names. The experiment is shown in chapter 3.

c) Assuming the availability of large amounts of quality annotated data, it is unknown if vanishing gradient problem will persist in the Morphological Rich Languages. Moreover, if a Morphological Rich Language has small and inadequate amount of annotated training data to show the expressiveness and variability of words in a language through positive and negative examples, it is assumed that deep learning methodologies will not be as performant.

2.7 Summary

This chapter covered the definition of named-entities in the literature and as they are explored in this dissertation. The general challenges of named entity recognition (NER) and state of the art approaches are discussed also. The next chapter explores Search and how named entities are difficult for search engines in English and MRL.

3 Search – Information Retrieval

One of the intellectual contributions of this research is to show the improvement in relevancy ranking of Search system as it is enhanced by named entities specifically for MRL languages. This chapter discusses what in general is meant by search, importance of named entities in search, challenges of correctly identifying names in state of art search engines in MRL and non-MRL, the background of search, and state of the art approaches and algorithms of search today in the commercial and in public domain.

The field of Information Retrieval (IR) or Search[7] involves finding disparate information in a corpus of data, usually documents. IR is the task of identifying and retrieving relevant documents from a set of data given a query as input. The user's information need is represented as a query which defines the **intent** of the information a user is seeking. The results are returned as a set of passages or documents that are **about** the intent of the query. This task[8] is usually performed by a computerized system. A text collection or a corpus may contain data in one language or in multiple languages, or in a language that is not known to the information seeker. There have been tremendous advances in the Information Retrieval (IR) community since the field emerged about forty years ago (Singal 2001) but, until the last decade, almost all automated research was done on English. Since then, the community has made some advances in other European and Asian languages like Spanish and Chinese, but still there has been little work on languages that are written in Arabic Script with an exception of Arabic. In the recent years there has been a strong surge of work done on Hindi in India, mostly in the area of natural language processing, not Information Retrieval.

With the explosive growth of data on the Web, there is a need to study and build language technologies on languages that, although spoken by a large number of the world's

[7] Information and Search are used interchangeably

[8] IR or Information Retrieval specifically talks about automated retrieval systems built to retrieve information instead of methodical manual methods for seeking information.

population, have not gotten attention in the search community primarily because of the lack of language resources like corpora (Baker 2003), (Riaz 2002). Arabic Script languages are an example of such languages. These languages are identified as resource challenged languages in the search community. Most of these languages were and still are transliterated on the Web or represented as images because of the lack of Unicode support of the Operating Systems, or the lack of support in the presentation software. For example, while trying to search for an original non-Latin script named entity transliterated in English, a query is submitted multiple times with various variants of spelling to retrieve adequate amount of relevant material. For example, Gita, the holy book of Hindu faith which can also be written as *Geeta*. Similarly, the holy book of the Islamic faith is written both as *Koran* and *Quran*. Almost universally, the news wires and the news stories are filed by the reporters in the field who are not properly trained to transliterate and transcribe on non-Latin characters in English, or are too short on time to look up the proper transcription and transliteration techniques. Therefore, the news stories are filled with many different spelling of a name. For example, consider some of the spelling from different news outlets for the late leader of Libya*: Qaddafi, Qadaffi*, *Qaddaffi, Gadafi, Gaddaffi and* Qazzaffi. This shows that correct name recognition plays an important role in search improvement. This lack of underperformance was also conformed as I studied the challenges faced by MRL languages to develop a NER system as described in section 4.5 and Chapter 5.

3.1 Named Entities in Search

Proper name recognition is one of the basic tasks in modern retrieval systems regardless of language (Lisbach 2015). Shalaan (2014) reports that up 71% of queries in modern IR system constitute named entities. For this research I analyzed 500 million query logs of a commercial search engine and discovered that half of the queries were about named entities specific to a domain. *Name screening* is a required and obligatory part of processes followed by banks, and financial industries before opening an account or doing business with a person or an organization. There are many commercial systems like World Check that specialize in this area. There are government agencies like OFAC in the US and

other countries across the world who maintain list of names of sanctioned entities. These entities are the names of individuals, organizations, and locations (mostly countries like North Korea or Iran). A system's goal is to correctly identify the named entity so an action can be made. The action could range from doing nothing to notifying law enforcement authorities like Interpol to deciding not to do business with that entity. This processing is called Identity Matching where names are matched against the database of known criminal, terrorist, politically exposed person, and other sanctioned parties. These days, these searches include international political parties, politicians, people associated with the politicians to combat corruption, and money laundering. For example, resolution of *John Fitzgerald Kennedy* and *John F. Kennedy* is not that trivial even in today's state of art technologies (Lisbach 2015).

3.1.1 Challenges of Names in Search

There are a number of challenges of name processing in search. These challenges can be represented in a spectrum if a search is performed on a MRL or non MRL. In this section, I will briefly describe two dominant challenges for transliteration and semantic ambiguity. Transliteration of one script to another is an issue for search regardless of MRL and non-MRL, but is compounded when a MRL Arabic Script language name is transliterated in English because a lot of valuable information for name resolution is lost. Therefore, better techniques are needed to search names in their original script.

Names of languages that are an MRL like Arabic, Persian, and Urdu are an example of such languages where a non-Latin script named entity transliterated in English does not retrieve all relevant results. For example, a query is submitted multiple times with various variants of spelling to retrieve relevant information. An example is one of world's most common first names – Mohammad, which can be spelled in English as at least a dozen different ways: Muhammad, Mohamed, and Mohd to list a few. However, Muhammad can only be written one way in Urdu, Farsi[9] and Arabic as محمد. The ambiguity that results

[9] Farsi and Persian are used interchangeably

in transliteration of names is also called the *Qaddafi problem*, named after the late ruler of Libya (Riaz 2007)(Round 2017)(Lisbac et al. 2015). For example, consider some of the spellings from different news outlets for the late leader of Libya*: Qaddafi, Qadaffi*, *Qaddaffi, Gadafi, Gaddaffi and* Qazzaffi. English transliteration to Qazzaffi is found in newspapers that originate from Urdu and Persian speaking regions because the late ruler is pronounced as */Qadafee/* Arabic and */Qazafee/*in Persian and Urdu.

Consider an example, where transliteration, transcription, and ambiguity of names are compounded. The holy month of fasting for people of the Muslim faith is written as رمضان and there is only way to write is the Arabic Script. News reporters writing a story about a religious event may decide to transcribe the month's name either in Urdu pronunciation *Ramzan* or Arabic pronunciation as *Ramadan,* and therefore may create ambiguity about the topic between documents, or in other words, increase the "distance" between documents that describe the same "concept". Besides Persian or Arabic based transliteration listed above, there are at least two other ways to transliterate the holy months name using Persian or Arabic phonetics.

Besides transliteration, to add complexity to name identification regardless of transliteration, there are many names in the Arab or Islamic world that are named after the Islamic months like *Rajab Ali* where Rajab is a month in the Islamic Hijri calendar. The mention of these names in a corpus addition to the Islamic month adds more ambiguity to understand the **about**-*ness* of the documents.

3.2 The Search Challenge

Besides the advances in search engine technology, searching today is keyword based. For example, Google is the state of the art for search engine technology today. If one searches for the word *car* on Google, there will be no documents found with the word *automobile.* Although this is a trivial example, with no important consequences, the result of not finding a relevant document could become very serious in a medical or in a legal domain. Consider an oft-referenced example from the legal domain, The McDonald's' hot coffee case where McDonald's was sued for serving coffee very hot. The attorney who is

representing either party will be interested in legal documents where some hot or cold beverage has caused an injury to a person. The attorney could be interested in the amount of major punitive damages that should be paid by a company of McDonald's size. If McDonald's is indexed as an organization, then more relevant documents will be returned. There is no publically available search engine that provides this functionality overtly. Let's refer to the search problem above as the Concept Search problem. There has been considerable research done in the IR community to solve this problem using the **bag of words approach** – where each word is in a document is treated independently without any notion of parts of speech, named entity, etc.

One of the biggest challenges of IR systems is that the documents retrieved as a result of user issuing a query are not ***about*** the intent of the user. In other words, the *concepts* in the query and the documents were not matched. Given this problem statement, this dissertation defines concept search to be **about-**ness of the query and further hypothesizes that **about-**ness can be increased by the enhancement of named entities.

Mismatch between the user intent and results problem manifests itself because of the query and document term mismatch. Such mismatch can happen for various reasons. One such reason is a language phenomenon called *synonymy*. Synonymy of a word means that there is more than one way to name and describe a concept within a language. Synonymy causes low recall and low precision. The other language phenomenon that hinders good performance of IR engines is *polysemy*. Polysemy means that two or more different but related concepts can be represented by the same term. An example of this phenomenon is a term like "chip" which can either mean a semiconductor chip or a casino chip in English and سحر/Sahr/ which could mean a magic spell, the morning dawn or name of a female person. Polysemy causes high recall, but poor precision. It is observed that people use the same keyword for a well-defined object less than less than 20% of the time (Deerwester 1990). In contrast to polysemy, *homonymy* means that the two terms have the same spelling and sound the same but have different meaning and therefore represent different words. For example, "bank" as a financial institution and "bank" as a

river bank. Homonyms degrade both precision and recall of the system. Languages have different levels of polysemy, synonymy, and homonymy. Usually languages that are more expressive and morphologically complex have high degree of synonymy, polysemy, and homonymy. Therefore, it is a greater chance to have mismatch between the **about-ness** of a query and terms during the search process.

The following sections describe approaches to solve the search challenge.

3.3 Keyword Based Search

The search process has a query and a set of documents or passages to be retrieved. Keyword-based engines are dominant and most widely used today. A keyword based engine scores and, for the most part, rank documents based upon the presence of a query term and could cause mismatch between the query term and the documents in the corpus. The mismatch between a query term and documents in the corpus manifests itself in two ways: 1) keyword search often retrieves a non-relevant document that contains a keyword e.g. plane (airplane, co-ordinate plane). Also 2) keyword search misses documents that are relevant but don't contain the query term. For example, a query term of coffee will retrieve documents about coffee not tea, but the information need was for documents about hot beverages. Keyword search is largely divided into four categories.

3.3.1 Boolean Keyword Searching

Boolean keyword searching is the basic approach used for searching online media. It is widely used in most commercial search engines. WESTLAW™ and LEXIS-NEXIS™ are examples where the search engine query can be a simple query of one word or a complex query with logical connectors. In such a query, only the documents that contain the terms that are listed in the query are retrieved. Contrary to the prevalent belief, this is a powerful mechanism where the recall and precision are very high when a user knows what they are doing and are experts with the query syntax. Boolean keyword search has some drawbacks in the hands of a novice user. Following is a list of drawbacks from (Moulinier 2002):

- *Large result sets:* A Boolean keyword query will result in all documents that match the query. This result set could be quite huge. Typically, a search session in Boolean query system involves refinement of the query again and again until a manageable size is returned for the query.

- *Complex query logic:* It takes some level of mastery of the query syntax for a particular search engine to remove the number of iterative processes. Although, the query syntax could be quite powerful, it could also be cumbersome. For example, consider this query from WESTLAW, a legal search engine. *Cat /s dog & sheep /p wolf & (injur! /4 person child).* This is a somewhat simple query from WESTLAW, which asks for the documents that have *cat* within the same sentence as *dog*, and *sheep* in the same paragraph as *wolf*, and finds all the morphological variants of the term *injur* which is four words apart from *person* or *child* on either side of the term *injur.* For another example, if a user wants to know **about** President Bush senior and his health, specific attention needs to be paid to craft a query such that that George W Bush is not included, and additional query terms like *Herbert* needs to be added. A query term can get very complex with special syntax. Appendix C has an example of a Boolean query that demonstrates trying to find information using named entities

- *Unordered result set:* The result sets are not ordered by relevance to the query. Instead, they are ordered by some other measure, such as the chronological order of the publication date of the document. This is a problem when one wants to find out the recent documents the Iraq war in 1991 and the documents returned pertain to the current Iraq war if a date restricted search is not part of the system.

- *Dichotomous Retrieval*: The results are the documents that exactly match the query, i.e. the documents that don't match the query are not returned. This could be a problem, when an overly complex query returns no documents. A fuzzy search will be helpful then.

- *Equal term weights*: All query terms are considered equally important, i.e. each query term is given equal weight. This is not a good strategy because a term like *Moqtada Sadr* (leader of the Iraqi Mahdi Army militia is more important in the query than the term *militia*.

  Modern commercial Boolean search engines like WESTLAW, Bloomberg BNA and LEXIS-NEXIS have a powerful Boolean engine where terms can be weighted.

3.3.2 Ranked Retrieval

Ranked Retrieval tries to overcome drawbacks of Boolean queries by using statistics on a document collection and query issued by a user. Ranked retrieval engines find documents according to the frequency distribution of the query terms within a document collection. If a document contains many occurrences of the query terms which are rather rare in a collection as whole, this suggests that the document might be highly relevant to the query term. The rarity of a term in a document collection was first introduced by Karen Spark Jones and is called Inverse Document Frequency *(idf_t)*. Term Frequency $\left(tf_{t,d}\right)$ is the frequency with which term occurs in a given document$d$. Document Frequency is the frequency of the term in the document collection. The weight of a term t in a document d is given by $w_{t,d} = tf_{t,d}.idf_t$. Document retrieval is accomplished by computing the similarity between the query vector q and the document vector d, using using the similary formula which like a cosine measure with terms that have weights.

$$Sim(q,d) = \frac{\sum_t w_{t,d} \ . w_{t,q}}{\sqrt{\sum_t w^2{}_{t,d}} \cdot \sqrt{\sum_t w^2{}_{t,q}}}$$

Equation 11

3.3.3 Probabilistic Retrieval

Probabilistic Retrieval tries to provide mathematical foundation to the ranked retrieval, since it is mostly based on powerful heuristics. The usual parameters of frequency and document counts are fed as the parameters of a Bayesian model to estimate how relevant a document is to a given query. InQuery from UMASS and its commercial cousin

WESTLAW's WIN system are notable examples. Probabilistic IR engines compute the probability that a document is relevant to the query. Probabilistic retrieval is based on the Probability of Ranking Principle which states that the ranking documents by decreasing probability of relevance to a query will yield optimal performance – that is, the best ordering according to the available data.

3.3.4 Vector Space Model – a variant of Ranked Retrieval

The Vector Space Model (VSM) dates back to the early 1960s and its use is popularized by Salton in 1975 when he used TF-IDF (Term Frequency and Inverse Document Frequency) as a weighting mechanism. Viewing Information Retrieval in the light of linear algebra and vector spaces was a novel approach that was first used in the SMART retrieval system developed at Cornell University. VSM represents a corpus as a term-document matrix (document-term matrix is also used) in a vector space. The idea is to represent a semantic space in which documents belonging to the same topic remain close to each other. In such a space, each keyword (term) is represented as a separate dimension. In other words, the dimensionality of the vector space is the number of unique terms in the corpus. The documents of the corpus are represented as the vectors in this space. Figure 5 illustrates keywords and documents. The query is represented as a vector in this space and is treated as a document vector. Typically, the document vector and the query vectors are normalized to unit length in order to give equal weight to small and large documents. The computation of similarity between the query and the documents is calculated using the standard dot product between the query vector and the document vector. The query is compared with each and every document vector, and the documents above a certain threshold are returned to the user. The similarity expression can be represented as:

$$Similarity\ (q, d) = \ q.d$$

$$= q^T d$$

Equation 12

There are other methods to compute the similarity between the query vector and the document vectors but dot product is the most convenient to explain. Typically, the vocabulary size of native speaker is about 50,000 words, but if one is eloquent and articulate, the vocabulary size could be about 100,000 words. So if we assume that the corpus contains about 1 million documents, the dimension of the term-document matrix will be of $10^6 \times 10^5$. This is a large matrix even to be processed by today's supercomputers (Belew 2000).

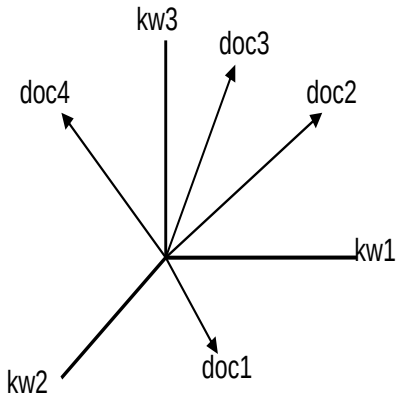


Figure 5

Below is an example borrowed from Story (1996) that best describes retrieval in a vector space model. We will use the same example to illustrate LSA state of art in traditional *concept search* based on dimensional reduction. In Figure 2, Doc-1 could be about animals found in Africa, and Doc-2 could be about plants and mentions animals that eat them.

	Apple	Animal	Zebra
Doc 1	0	3	7
Doc 2	10	1	2

Table 3

The similarity between Doc-1 and Doc-2 can be computed by taking the dot product of the Doc-1 and Doc-2 vectors. $0 \times 10 + 3 \times 1 + 7 \times 2 = 17$. The result of the computation is the similarity score. The vectors are not normalized to aid understandability. Suppose if a query is sent to the above system that contains apple twice because of stemming and zebra once, the query vector will look like the following:

	Apple	Animal	Zebra
Query	2	0	1

Table 4

Our system will score the similarity of Doc-1 to the query as 7, and the similarity of Doc-2 will be 22. Therefore, Doc-2 is judged closest in similarity to the query. If for some reason the query term contained the term *Honey Crisp* a type of apple instead of apple or the document talked about *honey crisp* instead of an apple, then a match will not be found.

3.3.5 Challenges of Keyword search

The traditional methods of retrieval fail to fulfill a user's information need. Deerwester (1990) presents three reasons for the failure.

- The first reason is *selective indexing*. This means that all terms that all users choose to use in a query are not all contained in the indices of the search engine. There have been a number of sophisticated methods employed for query expansion and use of taxonomy, but such methods, while trying to solve the problem of synonymy, introduce polysemy and homonymy.
- The second reason that these two methods fail is that there are no suitable methods to deal with polysemy and homonymy.
- The third reason is that words are treated independently of each other in each document. Two words that never occur together are weighted the same as two words that occur together most of the time. An example is that the terms *Iraq* and *Baghdad*

occurring together in a document are treated the same as the terms *Baghdadi* and *Madagascar* occurring together.

3.4 Traditional Concept Searching

Concept searching has been tried in the area of Information Retrieval time and time again to address the query and document mismatch problem with mixed results. The goal of traditional concept search is to reduce the gap of query terms and documents terms and not to directly to address the **about-ness** challenge. Traditionally, these techniques ranged from manual processing to sophisticated mathematical models. A few examples are: manual or automatic tagging of the words with categories and or senses, neural networks, ontology (taxonomy) based processing, dimensionality reduction (PCA, factor analysis, SVD), and cluster analysis.

One of the basic intellectual steps for this research is to understand the state of the art for concept searching in English and then compare the techniques to Urdu – an Arabic Script morphological rich language (MRL). The initial conjecture is that the translation of techniques to Urdu will not be smooth.

3.4.1 Latent Semantic Analysis

By far, the most studied technique for conceptual searching is Latent Semantic Analysis. In this section, the use of Latent Semantic Analysis[10] (LSA) for the task of concept searching is discussed. The heart of LSA is its use of Singular Value Decomposition. The use of singular value decomposition and its utility for concept searching is examined from the linear algebra perspective. In this section, LSA is examined as a special case for knowledge induction, Bayesian regression models, and term co-occurrence analysis, each explained in detail in appendix B. LSA and its probabilistic variant PLSA, have been baked into many commercial products.

[10] Latent Semantic Analysis (LSA) and Latent Semantic Indexing (LSI) are used interchangeably

The goal of concept search is to overcome the three problems mentioned by Deerwester (1990) to address the challenges of keyword-based search in Section 3.3.5. LSA tries to accomplish this goal by making an assumption that the terms represented in a document are unreliable and may be an incorrect representation of the document. These unreliable representations should be replaced by some other set of entities that are reliable representations of the document. Therefore, LSA tries to extract the higher order structures or latent structures in the association of terms and documents to extract such relationships (Deerwester et al. 1990) (Berry et al. 1994) (Landauer et al. 1998).

In almost all studies of LSA this analysis is done by using a term-by-document matrix using Singular Value Decomposition (SVD), which is a well-known method to perform quantitative analysis and dimensionality reduction on many different types of data sets (e.g. image analysis, eigenvector decomposition, and factor analysis). The essence of this section is taken from Deerwester, Berry, and Landauer. Each of them use different backgrounds to explain SVD; Deerwester uses factor analysis, Berry uses linear algebra, and Landauer uses cognitive and psychological experiments. In this section, LSA is explained using factor analysis and linear algebra terminologies.

Traditionally, one-factor analysis is done by the analysis of the matrix that represents associations of the same type of object among each other. This could be the document-document matrix (i.e., the entities on the rows and columns are the same). The analysis might show how closely the two documents are related to each other. The square document-document matrix is decomposed into the product of two special matrices through a process called *eigen-analysis.* These matrices contain eigenvector and eigenvalues. The most important realization is that these two new matrices show the breakdown of the original data into linearly independent components or factors. Usually, many of these factors are small and can be ignored. As a result, the new model contains fewer dimensions. Hence the similarity of the objects, in our case, documents, can be *approximated* by smaller number of factors or dimensions (Deerwester 1990).

Before proceeding further, it is important to understand the concept of eigenvectors because SVD decomposition results in matrices that contain eigenvectors and eigenvalues. We will not present the technical details here. Those can be obtained from a good linear algebra reference. One recommended reference is in Appendix A from Kumar et al. (2005). Eigenvectors are also called the characteristic vectors of the system. This means that an eigenvector is the vector that best describes the system in a particular direction. The eigenvalues are the scalar values that show the magnitude of that characteristic vector.

Returning to our factor-analysis, SVD is a two-factor analysis where the input is any rectangular $m \times n$ matrix (i.e., the square matrix condition is removed). Also, the rectangular matrix contains values that relate two different entities instead of one type of entity as in one-factor analysis. In other words, rows and columns represent two different entities. For IR it is a terms and documents and therefore, a $term \times document$ matrix. SVD decomposes this rectangular matrix into three very special matrices. The resulting matrices contain the singular values and singular vectors. Sometimes eigenvectors and eigenvalues are referred to as singular vectors and singular values, respectively. As with one-factor analysis, many of the components are too small to be of any significant value. Therefore, the new model has many fewer dimensions. This smaller model contains approximated values for the similarity of term-term, document-document, and term-document.

The dimensionality reduction while maintaining stability is a desirable property in information retrieval. As described earlier, a document is represented in a vector space as a vector of the documents constituent terms. Since the dimensions (keywords) are being collapsed or merged together by SVD, i.e., SVD replaces individual terms with the derived orthogonal factor values and, therefore, it brings the documents that belong to the same *concept* together. These factors or components are, in some sense, artificial concepts which represent the extracted common meaning components (constituents) of many different words or documents. Therefore, each term for the term-vector and

document for the document-vector is characterized by a vector of weights indicating strength of association with each of its underlying concepts. The original document vector is composed of the weighted sum of its constituent terms. Similarly, a document in a reduced space is also a weighted sum of its constituents in the reduced term-space. For example, if there are documents that contain the term *cars, automobiles, vehicles, elephants,* and *zoo,* and if the user's query contains *cars,* the documents with *automobile* or *vehicle* will not be returned. After reducing the dimension of the matrix, the query that contains the term *car* will also return the documents that contain the terms *automobiles* or *vehicles.* Similarly, the query about *zoo* will return documents about elephants.

One of the most critical aspects of SVD pertains to the number of reduced dimensions. Let that number be k. A value of k that does not reduce the dimensions so much that unrelated documents are mapped onto each other is desired. At the same time, all the related concepts need to come together. Equivalently, the meaning of a particular term or document can be expressed by *k* dimensions (factor values) or by the location of a vector in k-space defined by the *k* dimensions.

A query is treated in this reduced space as a pseudo-document representing a set of terms. These terms can be weighted also by a multitude of methods. So, a query is a weighted sum of its component terms. The query is mapped onto the reduced space and is compared to each and every document using cosine similarity (dot product). All the documents that are returned score higher than a preset threshold.
Alternate views of LSA are presented in appendix B

3.4.2 Commercial Search Engines and LSA

There are a number of commercially available search engines that claim to be using concept searching. A number of these search engines use LSA as their base along with some other technology. There are many reasons for using LSA as a starting point. Primarily, LSA is used because the engines do not have to do any syntactic parsing in order to build the indices. Also, LSA claims to be language independent because it is a "bag of words" approach and does not incorporate linguistic features and knowledge to generate

the concepts. Evaluation was done on two state-of-the-art commercial concept search engines, Reccomind™ and Engenium™. Reccomind™ uses Probabilistic Latent Semantic Analysis (PLSA), a novel form of LSA that incorporates probabilities, and Engenium™ incorporates LSA with some other non-disclosed technology. Neither software program is Web-based—they need to be installed on the user's site using the user's hardware, memory etc. because the computation required to calculate SVD is intensive. Both commercial engines were tried on a large collection of English data, but results showed unsatisfactory precision. The results are explained in the next section. No Urdu data on the commercial engines was tried on either system due to lack of support of the non-Latin characters.

3.4.3 Experiment on English and Urdu Data

As mentioned earlier, the most critical part for LSA, more accurately, SVD performance, is choosing the correct value of *k.* In most studies, (Deerwester et al. 1990, Berry et al. 1994, Landauer et al. 1998), the *k* values range from 80 to 100 to show the best precision and F-measure. The choosing of the *k* value is done by empirical methods and relevant judgments of human judges. The experiments on a large collection of legal data sets showed using commercial, and JAMA (matrix package for JAVA) showed that the 90% of the system's density was focused on the first 50 factors out of 3000 concepts. This reduced space included concepts that were severely morphed into each other. The query returned results that had low precision and recall (Riaz 2008). We acquired the code from the vendors and tuned it for our domain, but the results were still not satisfactory. For commercial engines, it is important to note that the dimensions of the initial matrix are unknown to the user, and the matrix is calculated internally according their proprietary methods after the data is ingested. This experiment suggested that for a specialty corpus of data, dimensionality reduction on documents negatively impacts the results.

Another experiment was to create and run the input matrix with the Urdu documents obtained from the Becker-Riaz corpus (Becker and Riaz, 2002). Since most of the experiments in the literature show good results, an initial space constructed of a small

number of documents that were related, but not obviously related. Polysemy and synonymy was introduced in the documents by adding words to the documents.

Below is the list of translated keywords for each document. The documents terms are translated also to show *what, where, and who.* The noise words are removed also

$Doc\ 1$: $\{prime\ minister, Shaukat, Citibank, Pakistan, Musharraf, leader, meeting\}$
$Doc\ 2$: $\{leader, America, Hindustan, Delhi, Sonia, country\}$
$Doc\ 3$: $\{Pathan, ball, wicket, cricket, Lahore\}$
$Doc\ 4$: $\{cricket, school, Shaukat, Lahore\}$
$Doc\ 5$: $\{Musharraf, Pearl, Sheikh, Bharat, parliament, Pakistan\}$
$Doc\ 6$: $\{Afghanistan, America, Sheikh, Al-Qaeda, Israel\}$
$Doc\ 7$: $\{Shaukat, medical, study, leader, school\}$

In the matrix *Hindustan* and *Bharat* both are the names of the country India and represent synonymy. *Shaukat* represents homonymy and polysemy in the document because it refers to both *Shaukat Aziz*, the former president of Pakistan, and *Shaukat Sultan*, a principal of Lahore's school. I introduced noise into the data by connecting Doc 7 and Doc 1 with the term *leader*.

The $cell(i, j)$ of the input matrix contained values 0 or 1, representing the presence or absence of the keyword. When the match for the query containing *prime minister* did not highly rank the document that contained the word *prime minister*, the matrix was modified to contain the frequency of the terms found in the document. Even after adding frequencies of the terms, results were not encouraging. The query vector contained only 1 or 0 for the presence or the absence of the term. The results started to look better when we started adding weights to the query. This is an interesting phenomenon because most of the literature does not mention what criteria are good for giving weights to the query in LSA. Frequency is a bad criterion for weighting the query, because, typically, the query terms are not repeated, but that criterion improved the results considerably. The improving of results by adding weight to the query terms was the inspiration to consider

named-entities as a query terms where these terms can be boosted when recognized and focused my research toward NER in Urdu to improve Search.

When VSM was used on unique term-rich queries, the results were much better and relevant document was always retrieved.

3.4.4 Analysis of Under Performance of LSA

In contrary to the seminal papers about LSA, results show that LSA does not fit the bill for true concept search. It is considered to be the de facto standard because it is somewhat easy to implement, it does not need to know the syntax of the language, and it has solid theoretical underpinnings. In recent experiments, when one moves away from the traditional term-document matrices, the results are better than the vector space model consistently, but by a very small amount—not 30% better as widely reported (Moldovan et al. 2005).

One of the reasons for poor performance could be that documents typically have a hierarchical structure of topics instead of a concept or blotchy concepts. LSA has the best chance of performing optimally when the corpus is a reasonably focused collection of meaningfully correlated document (Papadimitriou et al. 1998). Real data sets or corpora are not focused collections of text. Becker-Riaz corpus is constructed from news stories and by definition is not a focused collection of topics.

Our experiment showed that the concepts extracted are really latent. One cannot extract the meaningful concepts from the reduced matrix and examine them. This observation is similar to the back propagation technique of neural networks where weights cannot be interpreted.

One other reason for the poor performance of LSA could be because, after SVD computation, the decomposed matrices are orthonormal. This is a very useful property in image analysis, but this dependency hinders the coalescence of concepts because words and their usage in a document are not orthogonal across content. Although Deerwester et al. (1991) describe the removal of term independency as an advantage through

reduced space, the reduced space itself has independent vectors. This orthogonality constraint could be preventing LSA from extracting true concepts.

LSA claims to solve the polysemy problem by using the noise reduction property of SVD. This will only work when real meaning is close to the average meaning. Since an LSA term vector is just a weighted average of the different meaning of the term, when the real meaning differs from the average meaning, LSA will reduce the performance. An example of this phenomenon was seen in the Urdu example when *Shaukat* was representing two different named entities and when the Urdu word *mein* was used meaning both *in* and *I*. Hence, in a morphologically rich language like Urdu which has high polysemy and in a well distributed corpus LSA generally will not work.

Berry et al. (1994) reported that LSA will recognize names in the corpus. The results indicate that although LSA could recognize names, it will attach them to incorrect people. For example, it will incorrectly attach *Shaukat Aziz*, the prime minister to a school. In the NLP community, connecting the correct entities is called co-reference resolution. We hypothesize that because of diverse topics in the corpus, LSA is not a good tool for co-reference resolution.

Using LSA is an expensive proposition in terms of storage and computational issues. Real data sets are too large to be stored on today PCs or small systems. The term-document matrix is typically a sparse matrix, and, even after stemming and stop word removal, the term dimension can easily reach a half-million which adds to storage costs (Berry et al. 1994). Computationally, LSA requires the query to be compared to each and every document before returning the documents. When new documents become available, recalculating SVD is not practical and automatic folding-in of new documents can decrease the efficiency of the system. This approach is computationally expensive and also slow for any practical purposes. It is a hard sell when current search engines use inverted indices and return documents instantly.

Since LSA is a least-squares method, it is really designed for, and suited for, normally distributed data. Documents and terms are not distributed normally and, therefore, are not a good fit for least-squares method. The critique shows that LSA is not the panacea of the concept search, and it is a slight improvement over much simple vector space model.

Although many of the literature showed that LSA works well with toy examples and other corpora, it under-performed for the Urdu toy example. This could be because the descriptive capability of Urdu language is higher than English (Bashir, 2011) (Hussain, 2000)(Syeda, 2011).

LSA was originally utilized to increase the recall of search engines, but over time there has been attempts to achieve improvement in precision also.

3.5 Query Expansion

Query expansion techniques are often used to bridge the gap between query and the matching documents. Following is an example from Moulinier and Jackson (2002). Consider a query: *Who sells complete email solution for cell phones?* This query will not retrieve the document that contains the following sentence: *Gizmotron is a leading vendor of electronic messaging services for cellular devices.* One way to solve the above mismatch is to expand the query. This means to add terms to the original query that are related in "meaning" to the original query terms. In order to add new query terms we need some sort of thesaurus. There are two types of thesauri: Some are rudimentary representing synonyms and the other fairly sophisticated like WordNet (Miller 1995)

3.5.1 Use of Ontology / Thesaurus for Concept Search

At first, generalized query expansion is a good solution for concept search; it does not work as well because of the following reasons. Synonymy is not the only goal for concept search. Although *phone* is a *device*, device is a hyponym (hierarchical relationship) of *phone.* This is a difficult relationship to capture while creating thesauri. If such a relationship is not captured, then the precision will be too low – query will retrieve *computers* also. Homonymy and polysemy can be a problem also because the use of the word *cell* in the query can have two meanings, i.e. jail cell or cell phone or cell of a body.

Regional variants of the term usage can become a problem for example; the word *cell* means battery in my lexicon. Other examples are tube and subway, pop and soda, etc.

WordNet is a fairly sophisticated electronic thesaurus that is hand built by George Miller at Princeton (Miller, 1995)(Fellbaum, 1998). WordNet has a large vocabulary about 100,000 distinct word forms divided into lexical categories like nouns, adjectives, verbs etc. WordNet organizes words into synonyms sets called synsets; each synset represents a single sense of a word. These senses are organized in a hierarchical fashion representing taxonomy. This means that the term *vehicle* occurs above the term *car*. Research has shown that using WordNet alone usually does not improve the search functionality because of high recall and low precision (Madala et al. 98).

In many commercial research products that use ontologies, the concepts that are editorially created are used for partitioning the corpus into different sections for browsing the documents rather than improving the ranked list of the search results. The documents are tagged with the categories ahead of time during publishing of the corpus using machine learning or rule-based classification approaches. This approach requires significant time and subject matter expertise to create domain-related thesaurus, management of the hierarchy. For example, to create a thesaurus in the domain to Data Privacy, Data privacy officers, compliance officers, attorneys, and knowledge workers are needed to create an ontology, to create and then maintain the training data for automatic tagging of text using machine learning, or to create manual rules to identify the document. Such resources are very expensive to create and maintain.

Thesauri are considered linguistic resources and can be used as gazetteers for NER approaches also if specially created for proper nouns. Besides Arabic, there are no thesauri available for Arabic Script MRL. Currently, there is no known thesaurus publically *available* for Urdu – rudimentary or sophisticated like WordNet. There is one available for Hindi. But the Hindi WordNet is not suitable for Urdu research, since it contains parts of speech and metadata defined in Highbrow sanskritized Hindi. Also, most of the words it

contains are in highbrow sanskritized Hindi. Chapter 4.4 and Riaz (2012) offer detailed explanation of why Hindi resources are not suitable for Urdu research.

3.6 Graph Search or Entity-based Search

Recently another genre of search is emerging that is not general web search that generates ranked list based on query terms. Instead it focuses on Named Entities as they occur in news articles or social media sites like Facebook, Twitter etc. This type of search is touted as Semantic Search or cognitive search because it has the potential to do reasoning and Natural Language Processing (NLP). Facebook released Graph Search that lets its users find specific information about their *friends* (people) and their interests about *location* (geography), *businesses* (organizations) they are affiliated with etc. Alchemy API from IBM Watson services is another such search. In this type of search, a natural language query is provided and the result in answers to a question. For example, *"People who work at IBM and live in Austin and like Indian food"*. Terms like *People, IBM, Austin* and *Indian food* are all substitutable because they are stored in a *knowledge graph (KG).* This KG can be in a graph database like NEO4J, Big Data environment or a general purpose RDBMS. The search can be People-Focused, Business-Focused, Post-Focused (in a social media context). In essence, given a domain anything that is qualified to be an entity can be represented as a "node" in that domain's knowledge graph. For example, in a news context, network of news documents connected through named entities. In an academic domain, scholarly articles and their authors are the nodes. Co-occurrence of terms help derive connections that may either help in ranked list or mining of new information from the large seemingly unrelated corpus.

Named Entity Recognition (NER) is the underpinning of this kind of interesting, lucrative and emerging search techniques. There is no knowledge graph for Urdu that exists because there is no known high performing Urdu Named Entity Recognizer. Analysis of other Arabic Script languages like Arabic and Farsi shows that there is some work that exists to create the knowledge graph of those languages but it is done using the help of

Google Translate and use the English-based annotators like Alcheymy, Text Razor and DBPedia (Farhadi, 2014).

3.7 Evaluation in IR

Like TREC evaluation, we will use the standard IR measures of recall, precision, and F-measure to assess the performance of retrieval. Almost all the papers in IR research use user judgments as their criteria for success. Precision and recall are defined as follows:
Recall is a measure that determines the ability of the search engine to find the relevant documents. Technically, it is the ratio of the relevant documents retrieved to the number of relevant documents in the collection. It is usually denoted by R. Recall is calculated by the following expression:

$$R = \frac{|relevant \cap retrieved|}{|relevant|}$$

Precision is a measure that evaluates the ability of the search engine to find the documents that satisfy the user's need. Technically, it is the ratio of the relevant documents retrieved to the number of total documents retrieved. It is usually denoted by P. Precision is calculated by the following expression:

$$P = \frac{|relevant \cap retrieved|}{|retrieved|}$$

In practice, if a system shows high recall it has low precision and vice versa, this makes recall and precision competing measures. F1-measure is the harmonic mean of recall and precision in that it combines these concepts to give one measure for evaluation. It is usually denoted by F_1 and is expressed by the following expression:

$$F_1 = \frac{2PR}{P + R}$$

3.8 Summary

In this Chapter we saw that search is not a solved problem and it is specially challenging when a user's need about names. Polysemy and synonymy are major challenges for the

current search techniques. Variations in transliteration adds to the complexity in accurate identity resolution. Experiments in Urdu and English showed that adding weights to the query terms improves performance. Named entities are a logical choice to boost query terms in a news domain because they identify *who, when, where,* and *when* in a document. The next section describes the methodology used in this dissertation to show that NER-enhanced search boosts performance in MRL.

4 Methodology

This research demonstrates the effectiveness of named entity recognition on a range of MRL and non-MRL languages to improve search. Some MRL languages for research have a rich set of linguistic resources like corpora, stemmers, stop words, gazetteers, and some have very few resources. The languages chosen for this study are based on their morphological richness, distinction from each other, and most important the spectrum on the availability of linguistic resources. Arabic is a resource-rich MRL whereas Russian and Persian has some resources like a corpus and a stemmer. No resources are publically available for Urdu.

Since Urdu is a focus language of this dissertation among others, it needs basic enabling technologies and tools to do effective NER and Search. Section 4.1 describes that current resources from similar script MRL cannot be used for Urdu research. Section 4.2 dives into Urdu introduction and pertinent features of Urdu that are relevant and challenging for this research. Section 4.3 describes the creation of necessary enabling technologies to do NER-enhanced search. Section 4.4 describes the interesting relationship between Urdu and Hindi in the computational context. Section 4.5 describes the NER challenges in Urdu and details the Urdu NER algorithm created for this dissertation.

4.1 Resource Sharing among MRL

There has quite a bit of interest in the Arabic script language processing in the Natural Language Processing (NLP) community, specifically in the intelligence community and other organizations working for the government agencies. An example is Arabic Penn treebank from Language Development Consortium (LDC) (Maamouri et al. 2005). The number of Arabic linguistic resources like corpora, lexical analyzers, classifiers, and annotated material outnumbers other Arabic script languages. Arabic resources cannot be utilized for other Arabic script languages such as Urdu, Persian (Farsi), Dari, Sindhi, Punjabi, and Pashto which are spoken in the South Asian and therefore are considered low-resource languages. Arabic is a Semitic language, and other languages belong to the Indo-Iranian branch of the Indo-European languages (Schmidt 1999). Although Arabic

shares the same script and some vocabulary, their grammars and morphology are quite different. Moreover, among other Arabic Script languages, there are grammatical and orthographic differences. For example, Urdu has two letters, one is ٹ pronounced in IPA[11] as /ʈ/ and the other isت represented as /t/. In contrast, in Persian, only the letter ت /t/ exists. Similarly, in Sindhi alphabet, the aspiration is represented by one letter whereas in Urdu aspiration represented by two different letters, where the second letter represents the aspirated sound /h/. The above examples show that tools developed for one Arabic script language cannot be used out of the box without careful consideration.

Unlike Arabic and Persian, Urdu is an Indo-Aryan language akin to Hindi. Urdu shares it syntax with Hindi, but its morphology is quite intricate because it is a combination of many languages. This Sanskrit, Arabic, Persian, English and Turkish, to name a few (Schmidt 1999). For introductory teaching purposes, the phonological and grammatical levels can be used interchangeably, but at the lexical level Urdu has borrowed so much from Arabic and Persian, and Hindi from Sanskrit that in actual practice they have developed as different languages (Schmidt 1999). Urdu has loan words from all the above languages. These words follow the grammar of their original language. This usage of loan words is more than vocabulary usage but extends to morphology, and grammar also (Syeda 2011). Hence, Urdu is unique in features and linguistics aspects (Syed 2011) (Rahman 2006) (Schmidt 1999) (Davie 2009). Riaz (2012) describes in detail how Urdu and Hindi Non-Arabic Script MRL cannot share resources despite being syntactically similar but morphologically different.

Since Urdu is a resource lacking language, a number of resources need to be created for evaluation. These resources are required to do research in Urdu language and a major intellectual contribution of this research. The following sections describes the brief background of Urdu and details the features of Urdu relevant to this research and then

[11] International Phonetic Alphabet

describe enabling technologies that needed to be created to accomplish this research in Urdu.

4.2 Urdu

This section gives a brief introduction of Urdu, explains its orthography and grammatical features like morphology, and describes computational challenges of researching Urdu because of the orthography and grammatical properties.

Urdu is one of the major languages in the world, spoken by about 60.5 million people as their first language. Urdu is considered to be spoken and understood by about 300 million people all around the world, second only to Chinese (Rahman 2006). Urdu is the national language of Pakistan and one of the 23 national languages of India. It is also widely spoken in Nepal, the Gulf countries and in many enclaves in Brittan and North America because of a large number of the South Asian diaspora. These communities have their own newspapers, magazine, advertisement literature, etc. published in Urdu.

Urdu is mentioned in South Asian literature as early as 1200 A.D and sometimes referred to as *Raikhta* (ریختا). Urdu has many names in the literature, as Hindustani, Khari Boli, and Dakhni (Schmidt 1999)(Rahman, 2006). Urdu has many influencing languages like Persian, Arabic, Turkish, Sanskrit, and English among others. It is a lingua franca in Pakistan, and many other European and Middle Eastern countries with South Asian enclaves (Riaz, 2007).

Urdu, like all Arabic script language, is written from right to left for character but numbers are written left to right. Therefore, Urdu is essentially a bi-directional language. Although Urdu has Arabic script, its syntax is influenced by Hindi but its morphology is based on all the influencing languages like Arabic, Persian, Turkish, English and Hindi. This influence is most visible depending on the dialect (Schmidt 1999). Urdu has all the properties of a morphological rich languages. Furthermore, it has additional complexity because of the influence of other languages, which introduces significant polysemy, synonymy, and homonymy.

The intriguing aspect of Urdu's history is its rise, prevalence, and its acceptance as preferred language in the subcontinent (Gharvi 2013). Urdu in its Persian and Arabic influenced diction is called *Urdu Mualla,* and took root in the Imperial Mughal courts.

Urdu in its *Khari Boli* dialect was used in Sufi poetic traditions in the subcontinent to be accessible by the general public (Devine 2009)(Riaz 2007)(Bashir 2011). Annual elegiac readings in the Muslim month of Muharram is a tradition since 1400 years mostly in Arabic and Persian before Urdu's genesis. This epic elegiac poetry is considered the highest form of Urdu verse which was its pinnacle in 1800 and 1900s (Bailey 2008) (Ghuravi 2013). Poets narrated the epic battle Karbala by using Quranic, Vedic, and Persian references using loan words seamlessly to bring about strong emotional responses (Ghuravi 2013).

Persian has the most influence in Urdu. Schmidt (1999) best describes this as:

> *Urdu developed in close contact with Persian, which was the language of administration and education during the period of Muslim rule in India. Even after Urdu began to replace Persian as the language of poetry in the eighteenth century, Persian retained its official status for another century and remained a rich source of literary vocabulary in Urdu. Elements of Persian grammar have been borrowed along with the vocabulary, and knowledge of them is essential for reading literary Urdu, particularly poetry.*

The above discussion shows few aspects of the evolution of the complex morphology of Urdu, its polysemy, code-switching, its word order relaxation to address emphasis all of which makes automatic processing of language quite complex.

Urdu shows extreme register[12] variation that ranges from highly Persian influenced language of Urdu ghazals (a poetic form), and formal literature to highly anglicized Urdu of the upper class, and media. Although it is spoken by a huge population, many speakers of Urdu don't speak it natively. The native speakers of Urdu are known as *Ahl-e-Zaban (People of the language)*, an Arabic and Persian phrase. Urdu is taught as a mandatory

[12] A *register* of a language is a version of it that speaker uses for a particular situation. Chapter 6 mentions them as brows.

language in Pakistan sometimes in addition to the local language like (Sindhi, Pashto, Punjabi, Balochi, etc.). This causes variations in dialects and cause problems for linguists. For example, Punjabi speakers change the gender of nouns. For example,دہی /dahi/ *(yogurt)* is masculine in Urdu, but many people from Punjab and Sindh treat is as a feminine noun. Age plays a big role in the current register gap within Urdu speakers, where older generation tend to use high diction Urdu with Arabic and Persian vocabulary, while younger generation will freely use English words and even English phrases casually in regular conversation. This continual combining of more than one language by bilingual speakers within single utterances is called code-switching, and indicates *much more than vocabulary borrowing but a much higher linguistic influence* (Davies et al. 2009). Code-switching in particular is quite challenging while doing stemming and disambiguating anglicized names written in Urdu script and are causes of transliteration and transcription errors.

4.2.1 Orthography

The orthography of a language specifies a standard process of using the alphabets in a writing system. It identifies the rules about how the letters in the alphabet should be placed together to form a word or a phrase. The orthography of language has rules about graphemes – the smallest unit of the writing system, diacritics – accent marks, the relationship between a phoneme – the smallest unit of sound and a grapheme. Orthography also has rules about capitalization, numbers, and the punctuation marks to name a few.

4.2.1.1 Character set

Urdu alphabet has 38 letters, ten vowels, ten nasalized vowels, and 15 diacritic marks. Some authors have included the aspirated symbols as letters (Hardie 2003), but as a native speaker, I don't think that is the part of the original alphabet and should be ligatures. Although diacritics are an important part of the character set and necessary for disambiguation to understand the context, they are optional in Urdu.

The Urdu alphabet gets its influence from Arabic and Farsi but also has a number of letters inspired from Sanskrit. For example, ڑ /ɽ/ is quintessential Sanskrit sound. Urdu alphabet is an extended character set of Arabic and Persian. For example, Urdu has two letters, one is ٹ pronounced in IPA[13] as /ʈ/ and the other is ت represented as /t/. In contrast in Persian only the letter ت /t/ exists. Similarly, in Sindhi alphabet, the aspiration is represented by own letter Urdu is represented as two different letters where the second letter represents /h/. Complete Urdu letters with their IPA representation, Romanization form, and pronunciation is provided below. The first row is the letter, the second row is the pronunciation written in Urdu, third is the pronunciation of the letter in English, the fourth row is the Romanization of the phoneme, and the fifth row is the IPA representation.

[13] International Phonetic Alphabet

Figure 6

4.2.1.2 Ligatures

In a writing system a ligature occurs when two or more graphemes are joined together to form a new single letter or glyph. Urdu has a cursive script, written in a Nastaliq style which is different than Arabic or Persian. Some of the letters change form depending on the context where they appear during word creation. As letters are changing to form each glyph, these glyphs are either joiners or non-joiners. While writing, all characters join together to a shorter form until a non-joiner character appears or a writer stops because a word is formed. The joiner alphabets are called ligatures. A word can be composed of many ligatures. While writing on paper, there seems to a space between different ligature when there is none. This transformation is governed by the morphological rules for Urdu. For example three letters ع, ک, س (written left to right in order) when combined together

form عکس *Aks* meaning reflection. In this example, all the letters are joiners and there is only one ligature. Urdu characters have different shapes if they are the first letter, last letter or somewhere in the middle. Consider the same example with the letter ع which was transformed as عک as it combined with ک at the word initial position. The same letter will stand alone as word last position as presented as in the word pronounced as *shujah* meaning chivalrous or brave. The ligature system of Urdu can make stemmer development challenging for Urdu (Riaz 2007).

As a convention, the Urdu words are written in the Nastaliq style of Urdu script followed by the italicized transliteration and then the English italicized translation in parenthesis – the example above will be written as عکس */Aks/ (reflection)*.

4.2.2 Word Order

The standard word order for Urdu is Subject + Verb + Object (SVO) but depending on the context and specially to add emphasis, the word order can be relaxed. Urdu is both SVO and SVO and some rare situations a sentence construction of OSV and OVS is also permitted. Butt (1996) suggests that Urdu is a free-order non-configuration language. This behavior is very challenging during Parts of Speech (POS) tagging or Named Entity Recognition (NER). This is one of the reasons that POS taggers are unhelpful for NER in Urdu (Riaz 2010).

4.2.3 Vocabulary

Urdu has an exceptional capability of accepting vocabulary from the languages that influenced its genesis (Syeda 2011). It accepts Arabic, Persian and Sanskrit words for high diction or register, and English and Hindi for casual and light-hearted diction (Devine et al. 2009). Syeda (2011) suggests that the ability to accept a wide range of vocabularies from many languages enhances the expressiveness of a language. However, this feature of a language increases polysemy, synonymy, and homonymy. Schmidt (1999) calls these words loan words. The following table shows examples from some influence bearing languages. We have introduced polysemy, synonymy in some situations.

Language	Original script	Urdu loan word	Meaning	Variant in Urdu
Arabic	مغرب	مغرب	West	غرب
Sanskrit represented in Hindi	पश्चिम	پچھم	West	
English	Fridge	فریج	fridge	
Persian	قبر	قبر	grave	گور /Gor/, مزار /Mazar/
Turkish	Dunya	دنیا	world	عالم , جهان

Table 5

4.2.3.1 Izafat

Although Izafat could be discussed as a morphological phenomenon, it is considered one of the most challenging aspects of Urdu vocabulary for computational processing as discussed in NLP workshops and researchers. *Izafat* means 'to add' is a phenomena Urdu borrowed from Persian. It is an enclitic short vowel that joins two nouns or a noun and an adjective. It is often pronounced or written as -e- to combine two words. It can also be considered a modifier. It has two grammatical functions:

- *Noun-Izafat-Noun* to show the possessive relationship. For example, حکومت پاکستان */Hakumat-e-Pakistan/ (Government of Pakistan)*. This construction is the reverse of the possessive word order in Urdu پاکستان کی حکومت */Pakistan ki Hakumat/ (Pakistan's Government)*. The izafat construction is a proper noun referring to the Government of Pakistan in its official capacity and the non-izafat construction is common noun and refers to the entity ruling the country at this moment in time.
- *Noun-Izafat-Adjective* that shows the noun is modified by the following adjective. For example, وزیر اعظم /wazir-e-azam/ (Prime Minister) literally the great minister or عالم بالا /Alame-e-Bala/ *(Highest Heavens).*

The two nouns are connected with each other with a subscript called (zaer) but it is mostly unwritten in the electronic text because most computer systems don't support it. Identifying izafat words is one the most involved computational challenge for Urdu processing in our research for stemming and for Named Entity Recognition because there are no markers to identify them.

4.2.4 Parts of Speech

Urdu parts of speech are similar to English with a few additions. The Urdu parts of speech are noun, verb, adjective, adverb, pronoun, postposition, numeral, auxiliaries, conjunction, **haroof**, and **case markers** (Ijaz et al. 2007). Urdu has post positions, instead of English prepositions, but their function is the same. Haroof and case markers are additional parts of speech from English. *Haroof* are words that do not have a meaning of their own but when combined with another word form a meaning, e.g., *واہ/wah/ or* *نا/na/.* Some haroof when duplicated have semantic context also *نانا /nana/ (maternal grandfather)*. Case is a grammatical function of a noun or pronoun. Urdu has case markers which indicate the grammatical function of that noun.

4.2.5 Urdu Morphology

Morphology is the science and study of the smallest grammatical units of a language and formation of these grammatical units into words, including inflection, derivation, and composition (Davie et al. 2009). In other words, morphology is the grammar of the word and syntax is the grammar of the sentence.

Urdu has a complex morphology, and like Arabic, Persian, Hebrew, and Russian it is classified as a *morphologically rich language (MRL)* (Syeda 2011). The MRLs are the languages in which considerable information about the syntactic units and their relations is expressed at word-level, i.e., the structures of the words are complex, and morphological operations like inflection and derivation are more frequent (Tsarfaty et al. 2010). A precise description is provided in section 1.1.

Since the study of NER on MRL languages is part of this dissertation, some detail is provided in this section on morphology-related concepts with examples in Urdu and in

English for comparison. Morphological rules govern how a word can be made a plural by using inflection by adding an affix of some sort. Usually, in English a suffix -s is added to inflect a word to make it plural. For example, *pens* is inflected from *pen.* In Urdu فرد */fard/(person)* forms two inflected plurals one being افراد*/afraad/(people)* influenced from Persian. فرد */fard/* in Urdu exhibit polysemy with two other senses – *an official statement* and *unique*. A ***morpheme*** is the smallest constituent of a word that has a meaning. A morpheme explains a concept like a fan, a cloud or a relationship like in a word lifeless *less* is a morpheme. A morpheme in MRL can also express gender on a verb e.g. بھاگا */bhaga/ (he ran)* and بھاگی /bhagi/ *(she ran).*

There are many types of morphemes. For example, the *plural* morpheme construction in Urdu can be done many different ways and therefore it has many different ***allomorphs*.** The plural of the word کتاب*/Kitab/(book)* is کتابوں*/kitab + oon/(Books)* or another variation کتابیں*/Kitab+ain/ (Books).* For the word فرد */fard/(person),* the plural is formed using the Persian allomorph as افراد*/afraad/(people)*. **Free morphemes** don't have other constituents for a complete meaning. For example, بادل*/badal/(Cloud)* can stand on their own as a complete morpheme. **Bound morphemes** need to be added to other words to form a meaningful word, such as in English *pre* or *dis* etc. and in Urdu, the با /ba/(with) is used to form words like باکمال */ba+kamaal/(Superb)*.

Urdu and other MRL have a very active morphological process at the word level and therefore exhibit features like duplication, compounding, phrase creation, agglutinative nature, inflection, and derivation. It is essential to note that these mentioned features don't occur independently, i.e., a word or phrase may exhibit a number of these features. These characteristics introduce synonymy, polysemy, metonymy, homonymy and other morphological variations per the previous discussion. The MRL languages pose some challenges for machine learning, natural language processing, named entity recognition, machine translation, and other information processing areas because there are frequent misalignments while discovering the correct **about-ness** of the information between the indented concept and the retrieved concept.

4.2.6 Challenges of Urdu Processing

As Urdu is utilized and spoken by a huge population around the world, there is a great need to have computational systems that can address its orthography, linguistics characteristics for text processing. The first known attempt to process Urdu computationally was to explore Named Entity Recognition by Becker & Riaz (Becker et. al. 2002) that was abandoned because of no linguistic resources and resulted in the construction of the Becker-Riaz corpus (Riaz et al. 2002). Computational processing of Urdu requires both linguistic knowledge, and knowledge of computational aspects like character encoding on different operating systems to name one. Some examples of challenges in Urdu processing are provided below.

- Complex morphology and use of derivation and inflection makes stemming[14] arduous, e.g., the use of Izzafat from Persian.
- Optional use of diacritics which confuses value word vs. a stop word.
- Free word order, It both Subject Object Verb (SOV), and Subject Verb Object (SVO).
- Acceptable variance and acceptability in syntax when emphasis is needed.
- The use of Arabic Script in Nastaliq makes boundary detection a challenge.
- Unusual amount of code-switching.

This section describes the challenges in Urdu computational processing. These challenges pertain to the issues I have run into during this research and do not discuss all grammatical and engineering challenges in all areas. For example, syntax related challenges while creating a POS tagger are not explored as part of this dissertation.

At a high level, the following challenges can be categorized as below:

- Language properties and complexity
- Engineering challenges to process non-Latin script
- Lack of linguistic resources or enabling technologies to do research

[14] A process to conflate two words together to aid in search. Stemming will be explained in detail in the stemming section.

4.2.6.1 Language properties and complexity

4.2.6.1.1 Inflection

Inflection process of language entails changing the form of the word to express a different grammatical category like forming of plurals. Urdu is a highly inflected language because the loan words are inflected according to the grammar of the language they are loaned from. For example, Arabic loan words are made plurals according to the Arabic grammar, e.g., عالم*/Alem/(world)* → عالمین */Alem+een/(worlds)* whereas Persian words are made plural according to the Persian grammar e.g., فرد */fard/(person)* → افراد */afraad/(people)*. This challenge manifests itself while writing a stemming algorithm.

4.2.6.1.2 Derivation

Derivation entails new word formation by adding another word or some conflated stem of another word as an affix. The new word many times has a new part of speech or a new meaning. For example, نور*/noor/(heavenly glow)* → نورجہاں */noor+jahan/(beautiful)*.This characteristic poses a challenge for stemming, stop word identification, and named NER because the addition of new word can change an adjective to a proper noun. For example, نورجہاں */noor+jahan/(beautiful)* is a name of a famous singer in the Indian subcontinent.

4.2.6.1.3 Phrase Creation

The phrase creation rules of Urdu are complex as much as the morphology. Although I have not found much literature directly addressing the issue, one can infer the complexity in the syntax and morphology research review. Phrase identification effects are quite important for indexing to understand the *what* aspect of ***about-ness*.**

- Combination of two words to form another word is called compounding, and in a strict sense is not a phrase in English, but in Urdu it is treated as phrase with a space between the two words in electronic representation. For example, نمک */namak/(salt)* and حلال */halal/kosher/* → نمک حلال */namak halal/(loyal)*. In this case, the new word or electronically represented phrase in Urdu has changed its meaning. This phenomenon is also available in English as *blackboard.*

- The phrase can also be generated when two different words are combined with a *connector.* In this case, the original meaning of the word each word is retained but the phrase meaning is related but not the same. Consider a phrase جان و مال /jaan-o-maal/(wellbeing) where جان /jaan/life is connected with a connector و /o/(and) to مال /maal/(wellbeing) to form a new phrase. Special rules need to be looked for stop word removal, search, NER, and stemming to preserve the phrase's identity. Retrieval of this particular pattern was discovered by chance during experimentation and analysis on Urdu corpus and is explained in chapter 5. Since و is a high IDF term, its presence in a phrase brings back documents that have this pattern.
- Izafat is another form to create a phase which is borrowed for Persian. Details of Izzafat are provided in section 4.2.6 in this chapter. Most computational systems ignore processing it because of optional diacritics.

4.2.6.1.4 Duplication

Duplication of words is common in Urdu for emphasis. The challenge is how to distinguish if the duplication is done for emphasis or if it is a proper noun or a common noun. Given a word, it could be bound morpheme, like بی */bee/(bound morpheme)* or بی بی */bee-bee/(a woman)* or part of proper noun used for emphasis or as بی بی سی */bee-bee-cee/(BBC)* which is a proper noun and named entity.

4.2.6.1.5 Intensification of verbs

In order to make a compound verb, a root verb is combined with an intensifying verb to provide more emphasis or intensify the verb (Schmidt 1999). The intensifying verb can be put anywhere as an infix, suffix or in a postfix of the verb depending upon the root verb. Consider the root verb مار/maar/(hit) combined with the intensifying verb ڈالا/dala/(put) forms مارڈالا/maar+dala/(murder). This particular feature affects effective stop word identification because many times the intensifying word is considered as a separate word and is a candidate for "non-searchable" term because of its length or it frequent occurrence in the corpus. For example, the word لو /lo/(take) is an intensifying verb and

also a good candidate for stop word. The heuristic of stemming does not stem it as it is only two letter Urdu word.

4.2.6.1.6 Lexicon

Urdu lexicon has many of its influencing languages as shown in the vocabulary section and has many different registers. The richness or the diversity of lexicon makes computational processing for stemming, and the recognition of proper nouns quite complex. A complex morphology example can be illustrated by the following example, Urdu words one is formed by a bound morpheme با→باکمال*/ba-kamaal/(Superb)* and the other a free morpheme بادشاہ*/badshaha/(king)*. A stemmer for search will like to remove the bound morpheme با , and NER system will not want to because it is it is part of the cue word.

4.2.6.1.7 Code-switching

Code-switching indicates much more than vocabulary borrowing. Instead, it is higher linguistic influence (Davies 2009). Code-switching is when bilingual speakers use more than one language in a single utterance or in a sentence. Besides speech, code-switching is observed in Urdu writing. During speech, the code switching happens with all the influencing languages depending upon the register. In written text, the frequency of code-switching is mostly from Urdu to English where English is transliterated in Arabic Script. Occasionally, a Latin script word is inserted into Arabic script. The insertion of Devanagari script in Arabic script is non-existent. It is not uncommon to see a right to left flow of Urdu interrupted by a word written in English orthography and then the continuation of the flow right to left. For example, وہ میرا laptop ہے [That is my Laptop]. In this example, the Microsoft Word did not support English embedding within the Urdu sentence and displayed it improperly. Due to code-switching, named entity recognition and co-reference resolution become quite challenging when sometimes a name is written is its English spelling and at other times in Urdu transliteration. A notable example is *Guantanamo bay* and گوانٹاناموبے where both terms refer to the same thing. Moreover, the English representation is a phrase whereas the Urdu one is a word (Riaz 2010).

4.2.6.1.8 Word order

Although Urdu is considered a Subject + Object + Verb language (SOV), SVO and other variations are also allowed. Butt (1995) suggests that Urdu is a free word order language. Free-word order can cause serious issues when there is polysemy, and homonymy for proper noun identification. The reason of free word is Izafat, and case markers that are recognized as parts of speech and help deduce the right meaning of object and subject.

There are other challenging features of Urdu that are subtasks in this research. Those challenges will be explored in their respective sections. For example, NER related challenges like agglutinative nature of Urdu will be explored during the Urdu NER discussion in section 4.6

4.2.6.2 Language Engineering issues for non-Latin script

Urdu is a right-to-left Arabic Script language where its alphabets and glyphs are not supported in in the ASCII character set. Such languages are represented with character sets encoded in Unicode. The discussion and details of Unicode are out of the scope of this proposal but for more information see http://www.unicode.org. Since most of the language processing tools are not Unicode compliant, new tools need to be developed. Although the full support of Unicode storage is standard in modern systems today, the rendering software does not correctly support the display in right to left mode. For example, in the latest release of Elastic Kibana – state of the Art search and logging visualization software – Arabic character rendering is not supported. To process Urdu, familiarity with the Arabic Orthography and Unicode Standards is very helpful to debug and understand if the information is represented correctly. For example, EMILLE corpus is riddled with incorrect vowel representations (Riaz 2010).

Not all programming languages support Unicode and bidirectional languages. Java and C# in the .NET framework are the two languages widely used for Unicode processing. Operating Systems platform needs to have Unicode support built in. For example, Windows 10 has this support built in only at the Ultimate and Enterprise level. Not all

versions of Linux support Unicode. The latest Mac Outlook 2016 from Microsoft does not support the Urdu language although it supports Unicode.

To conveniently write programs, debugging and quick prototyping, a good Integrated Development Environment (IDE) is a necessity. Although a programming language like Java supports Unicode, Eclipse (the most widely used Java IDE) does not support Unicode in its display console debug statements that need to be written in a Unicode enabled file instead of simple *print* statement on the console. A text editor is also required that supports Unicode and Urdu fonts to view Urdu characters. Not all fonts support Urdu characters, and indeed, an editor that supports Arabic does not necessarily support Urdu. The lack of support is because Urdu has letters that not part of Arabic or Farsi. For example, the letter corresponding to /ڑ/ in *larddki (girl)* is not part of either Arabic or Farsi, it is a phonetic sound in Tamil, a Sanskrit-based south Indian language. Most free text processing editors don't support Urdu. Most of the editing of this research was done in Microsoft Visual Studio in editing mode.

We encountered considerable amount of font and code page issues during programming and also while storing data in the corpus. Besides storing the data, the correct fonts needed to be used in order to get the correct representation of the Urdu character. Some of these challenges are described during the corpus construction discussion.

The lack of linguistic resources challenge is discussed in the enabling technologies chapter.

4.2.7 Summary of Urdu discussion

This section described Urdu's history, its orthography as the only Indo Aryan language to have Arabic Script, its complex morphology because of influencing languages, and its pervasive borrowing including variant registers. All these properties make Urdu computational processing quite complex and show that its descriptive capability is higher than other languages (Hussain 2010)(Bashir 2011)(Syeda 2010). Also, the challenges faced while processing an Arabic script non-Latin language make researching in Urdu a very

time-consuming task because of encoding, no set “qwerty” keyboard and lacking of support in presentation software.

4.3 Enabling Technologies

In order to do language processing tasks like natural language processing, information retrieval, machine translation and named entity recognition, there are a number of digital linguistic resources required. For example, a corpus for data in the original language, a stemmer to conflate terms, a stop word list to remove non-functional words, parts of speech taggers, a set of judgments for baseline, and dictionaries for lookups to name a few. For most of the machine learning tasks, one needs an annotated corpus for training and test set.

The lack of digital resources presents a formidable obstacle for NLP in morphologically rich languages in general and specifically for NER (Shalaan 2014). Study of NER of other morphological rich languages like Arabic, Persian, and Russian confirmed the necessity of such resources and explained in detail in section 4.5. Unfortunately, when we started our work in Urdu language processing, none of these resources existed (Becker-Riaz et al. 2002). Although there is trickling of resources like POS tagger, keyboards, most of them are focused towards language feature understanding instead of machine learning or information retrieval (Riaz 2002)(Hardie 2003) (Hussain 2005) (Butt 2002). Moreover, if there are some resources available, they are for commercial use. There are numerous quality and coverage issues for the little resources available (Riaz 2007)(Riaz 2010)(Riaz 2012).

4.3.1 Corpus Construction

For information retrieval, NER and other related tasks some data are required. Usually, this set of documents is annotated from one or many collections called a corpus. A good corpus needs to somewhat balanced about the distribution of topics. A set of news stories over time is a good example of such corpus because of variations in topics. Domain adaptation task requires a somewhat focused corpus about the domain, but it still requires a good distribution of topics.

Currently, there are only two known Urdu corpora available to the community. One is the EMILLE Lancaster Corpus, in which Urdu is one language among many, and is the more

comprehensive of the two (W0038 2003) only in the number of tokens. The other is the Becker-Riaz Urdu Corpus, a corpus of strictly BBC Urdu news articles (Becker and Riaz 2002). The Becker-Riaz corpus and the EMMILLE corpus (Hardie 2003) were created roughly at the same time without knowledge of each other. Becker-Riaz corpus was created to study the exploration of NER in Urdu (Becker and Riaz 2002) and concluded that there is no corpus available. This led to creation of a general purpose corpus for language processing and information retrieval (Becker and Riaz 2002).

The lack of Urdu language corpus is due in part to online Urdu newspapers publishing in graphics instead of text, a practice which makes compiling a corpus of those online newspapers time-consuming and expensive (Becker and Riaz 2002). When inquired one of the Urdu newspapers in Lahore, Pakistan for electronic text formats of their news articles, but found out they do not electronically archive the news stories because they are not considered valuable.

The choice to publish in graphics makes it difficult for data harvesters to snag data from the Web. If one wants the data that are published in graphic form, one has to rekey the text, scan it using optical character recognition technology, or contact the publisher for electronic copies of text. Once the data in hand, one needs to be able to handle or convert from the character set in which the text was originally typed.

Both the Urdu corpora use the Corpus Encoding Standard (CES) to mark up the corpus, retain Urdu's Arabic script and are stored in Unicode—three characteristics that make either corpus desirable to use for digital processing (Ide et al. 2000). Both corpora are quite different in their characteristics. The differences between them explain some of the differences in the results and the rationale to pick one over the other for a given task. The Becker-Riaz corpus consists of approximately 7000 documents. The EMILLE corpus for the monolingual written Urdu documents consists of 368 documents. The documents in the Becker-Riaz corpus are small news articles of varying sizes from BBC Urdu (British Broadcasting Company). The documents in the EMILLE corpus are quite large and discuss one topic in detail. Some of the topics span documents. It is presumed that the EMILLE

corpus was transcribed or re-keyed. The reason for such a hypothesis is the prevalence of typographical errors. For example چےزوں (*things)* should be written as چیزوں. The error is the incorrect use of *yeh*, which should be ي instead of ے. The above explanation is by no means a criticism of the EMILLE corpus. It is mentioned to highlight one of the side effects of typographical errors that impact tokenization and therefore could impact some aspects of the algorithm. In this case, it will affect stemming and the case marking or gender of a noun. Also in this case, the word with the error and the correct word will be considered two different words in the corpus by the tokenizer.

Two native speakers did an analysis of the EMILLE corpus showed that there are a number of typographical errors, the case markers are incorrectly placed, a number of spelling problems where two words are incorrectly combined to form a single word and vice versa. These observations show that this corpus is not fit for purpose for NER and Information retrieval, stemming, but it is a good candidate for stopword generation. It is not a good candidate for NER because of the errors in case markers, and the lack of named entities in long documents. For retrieval, the total documents in the corpus were small and individual documents were long so that a query would have retrieved most documents. In contrast, Becker-Riaz corpus is fit to purpose for NER and retrieval. Both EMILLE and Becker-Riaz corpus follow Corpus Encoding Standard (CES) to create the corpus.

A document form Becker-Riaz corpus in CES is shown below:

Example of a document of Becker-Riaz corpus

```
<?xml version="1.0" encoding="utf-8"?>
<cesDoc version="2.0">
  <cesHeader type="text" creator="Kashif Riaz" version="2.0" status="update" date.created="04/29/03"
date.updated="05/20/04" lang="ur">
    <fileDesc>
      <titleStmt>
        <h.title>020716_sasia_media</h.title>
        <respStmt>
          <respName>BBC Urdu Service</respName>
        </respStmt>
      </titleStmt>
      <editionStmt version="2.0" />
```

```
  <extent>
    <wordCount>157</wordCount>
    <byteCount units="kb" />
  </extent>
  <publicationStmt>
    <distributor>BBC Urdu Service</distributor>
    <pubAddress>PO Box 3101, Islamabad, Pakistan</pubAddress>
    <eAddress type="email">urdu@bbc.co.uk</eAddress>
    <eAddress type="www">http://www.bbc.co.uk/urdu</eAddress>
    <availability status="unknown">© BBC MMIII</availability>
    <pubDate value="2002-07-16 13:13:00">07/16/2002</pubDate>
  </publicationStmt>
  <sourceDesc>
    <biblStruct>
      <monogr>
        <h.title>020716_sasia_media</h.title>
        <h.author />
        <imprint>
          <publisher type="org">BBC Urdu Service</publisher>
          <pubDate value="2002-07-16 13:13:00">07/16/2002</pubDate>
        </imprint>
      </monogr>
    </biblStruct>
  </sourceDesc>
</fileDesc>
<profileDesc>
  <creation date="04/29/03" />
  <langUsage>
    <language iso639="ur">Urdu</language>
  </langUsage>
  <wsdUsage>
    <writingSystem>Urdu Naskh Asiatype</writingSystem>
  </wsdUsage>
  <textClass>
    <catRef target="ONA" />
    <h.keywords>
      <keyTerm>Urdu</keyTerm>
      <keyTerm>Pakistan</keyTerm>
      <keyTerm>pakistan</keyTerm>
      <keyTerm>india</keyTerm>
      <keyTerm>bangladesh</keyTerm>
      <keyTerm>nepal</keyTerm>
      <keyTerm>press</keyTerm>
      <keyTerm>tehelka</keyTerm>
```

```
        <keyTerm>shahin sehbai</keyTerm>
        <keyTerm>tufail</keyTerm>
        <keyTerm>bbc</keyTerm>
        <keyTerm>urdu</keyTerm>
        <keyTerm>reporters sans frontiers</keyTerm>
        <keyTerm>rsf</keyTerm>
        <keyTerm>maoists</keyTerm>
       </h.keywords>
     </textClass>
   </profileDesc>
   <revisionDesc>
    <change>
     <changeDate>05/20/04</changeDate>
     <respName>Kashif Riaz</respName>
     <h.item>Replaced Doc n with the original document title as stated in either the BBC HTML header or
the file name. Captured the keywords from the original BBC Web page.</h.item>
    </change>
   </revisionDesc>
  </cesHeader>
  <text>
   <body>
    <p>
     <title>جنوبي ايشيا: پريس
        ميں زنجيروں</title>
    </p>
    <p>تحرير: طفيل احمد، بي بي
        سي اردو آن لائن</p>
    <p>پرل کے اغوا اور قتل امريکي صحافي ڈينيئل
        گيارہ ستمبر کے حملوں کے کے مقدمے کے فيصلہ کے تناظر ميں دنيا کي نظريں
        صحافيوں کي مشکلات پر مرکوز ہوئي بعد جنوبي ايشيا ميں کام کرنے والے
        ہيں?</p>
    <p>ليکن پرل کيس اپني نوعيت کا واحد واقعہ نہيں ہے? پاکستان ،
        اب ہندوستان، نيپال اور بنگلہ ديش ميں ميڈيا کو گذشتہ چند دہائيوں ميں
        تک کے سب سے بڑے بحران سے گزرنا پڑرہا ہے?</p>
    <p>ان ملکوں ميں صحافيوں
        سياسي انتقام، دہشت گردي مخالف فوجي پر حملے شدت پسند مذہبي رجحانات،
        خلاف مہم کا نتيجہ ہيں کارروائيوں يا ماؤ نوازوں کے?</p>
    <p>ميں کام پاکستان ميں فوجي حکومت کے دور
        برقرار رہتا ہے کرنے والے صحافيوں پر بھي بالواسطہ طور پر فوج کا دباؤ?
        فرانسيسي ادارے چند ماہ قبل صحافيوں کے حقوق کا دفاع کرنے والے
        شائع رپورٹرس ويداؤٹ بورڈرس نے اسلام آباد سے</p>
    <p>نيپال: ماؤنواز مخالف کارروائي جاري
          ہے</p>
   </body>
```

```
  </text>
</cesDoc>
```

Note that this document is **about** a number of topics. These topics are represented in English by the publisher in the XML by the keyword element. These topics they are also named of named entities. This annotation of named entities as topics further strengthens the claim of this thesis that the named entities play a critical role in determining the **about**-ness of a document.

4.3.1.1 Corpus Encoding Standard (CES)

The corpus encoding standard is a corpus creation standard in XML. An XML format provides needed standardization so that a user, who is unfamiliar with the corpus data, but familiar with a given XML DTD, can interface with the corpus fairly efficiently. Software that has been previously designed to handle a corpus marked up in a given CES XML structure can handle a new corpus marked up in the same structure. This is advantageous because someone does not have to comb through the new corpus trying to understand its design to redesign the software that interfaces with the corpus. The designer of a corpus is always familiar with their own design, so one advantage of using a standard schema of a corpus is to make the corpus readily available to other researchers. Corpus Encoding Standard (CES) was selected XML DTD to mark up our corpus (Ide et al. 2000). The main enclosing tag in this DTD is <cesCorpus> which is broken into main parts, <cesHeader> and <cesDoc>.

The header <cesHeader> contains meta information about the corpus data such as date created, creator's name and contact information, description of the source, categories of the content, the writing system of the language being stored, how hyphenation in the source text is handled, and much more information.

```
<cesHeader type="corpus" creator="Dara Becker-Kashif Riaz" version="1.0" status="update"
 date.created="2/2/02" date.updated="4/17/02">
 <fileDesc>
  <titleStmt><h.title>Urdu Corpus</h.title></titleStmt>
  <editionStmt version="1.0a"/>
```

```
  <publicationStmt>
   <distributor>Dara Becker&Kashif Riaz</distributor>
   <telephone></telephone>
   <eAddress type="email">riaz@cs.umn.edu</eAddress>
   <eAddress type="www">http://www.cs.umn.edu/~riaz</eAddress>
   <availability status="free"/>
  </publicationStmt>
 </fileDesc>
</cesHeader>
```

The document tag <cesDoc> is where the actual text of the language of interest is stored. Each document is itself marked up with metadata specific to each document, like topic and source information. The language data inside the <cesDoc> tags can be marked up simply with a paragraph tag <p> (Figure 2) or they can be more elaborately marked up with tags of semantic value (e.g., date, number, measure, name, term, time, foreign word) and formatting value (e.g., figure, table, p, sp, div, caption). Tags that indicate formatting features such as "caption" are important because they can be used, for example, to automatically determine the topic of a story.

```
<cesDoc>
 <title>جنوبی ایشیا: پریس زنجیروں میں</title>
    </p>
 <p>تحریر: طفیل احمد، بی بی سی اردو آن لائن</p>
 <p> امریکی صحافی ڈینیئل پرل کے اغوا اور قتل کے مقدمے کے فیصلہ کے تناظر میں دنیا کی نظریں گیارہ ستمبر کے حملوں کے بعد جنوبی
      ایشیا میں کام کرنے والے صحافیوں کی مشکلات پر مرکوز ہوئی ہیں۔</p>
 <p> لیکن پرل کیس اپنی نوعیت کا واحد واقعہ نہیں ہے۔ پاکستان ، ہندوستان، نیپال اور بنگلہ دیش میں میڈیا کو گذشتہ چند دہائیوں میں اب تک
      کے سب سے بڑے بحران سے گزرنا پڑرہا ہے۔</p>
 <p> ان ملکوں میں صحافیوں پر حملے شدت پسند مذہبی رجحانات، سیاسی انتقام، دہشت گردی مخالف فوجی کارروائیوں یا ماؤ نوازوں کے
      خلاف مہم کا نتیجہ ہیں۔</p>
 <p> پاکستان میں فوجی حکومت کے دور میں کام کرنے والے صحافیوں پر بھی بالواسطہ طور پر فوج کا دباؤ برقرار رہتا ہے۔ چند ماہ قبل
      صحافیوں کے حقوق کا دفاع کرنے والے فرانسیسی ادارے رپورٹرس ویداؤٹ بورڈرس نے اسلام آباد سے شائع</p>
 <p>نیپال: ماؤنواز مخالف کارروائی جاری ہے</p>
    </body>
    </text>
    </cesDoc>
```

4.3.1.2 Corpus Creation Process

The goal for Urdu corpus creating was to investigate Urdu related Natural Language Processing and Information Retrieval tasks like NER, stemming, etc. For such tasks a general purpose and named entity rich corpus needed to be created which is not focused. A news-related domain is suitable for that purpose because it contains a number of named entities and various topics. The corpus creation system included a web crawler to mine the stories and transformed the HTML using XML aided parser using Java programming language. Since the BBC stories are filed from all over the world with different reporters, a number of variations in the HTML structure existed. Also, the metadata was populated with author's name instead of the topic and many other inconsistencies. To remedy the inconsistencies in the data, a User Interface was created to see each document with all the metadata and story, and manual corrections were done using the input method described in next section. The corrected document was added to the corpus. Over time, if we found general patterns, they were accommodated into the program.

- **Urdu Input Method**

The Urdu corpus is developed over a number of years. These days some of the Urdu input methods are used from Microsoft Urdu Keyboard layout. In earlier days significant language engineering was used to input the documents in the corpus. Some earlier day challenges included using Arabic-based input system, but they did not include some Urdu specific letters. To remedy it some Persian specific keyboard letters were used. There were still some letters that were not found in Persian and Arabic and therefore MS symbols were used input them into the corpus

- **Fonts**

Storage and processing of Urdu data in the Unicode character set eliminated significant number of graphical and image issues. However, a number of font issues had to be resolved because there is no standard for mapping Unicode-based fonts to the Arabic subset of Unicode. For example, the Urdu letter ہ /hey/ has a number of variations as

shown in table. For rendering, Urdu Naskh Asiatype was used which was available from the BBC Urdu Website and incorporated in the code to process content correctly.

How Fonts display in variation of letters *heh*ہ

*because of inability of this version of Windows OS, some Urdu variations of letter are not shown

Variation of *heh*	Urdu Naksh Asia Type display	Arial Unicode MS display
06C1	FBA6 ہ or FEE9 ہ	FBA8
	FEEA	FBA9
	FBA8	
	FBA9 ہ	
06BE	FBAA ھ or FEEB	FBAA ھ or FEEB
ھ	FEECھ	
0647	Not found in the corpus	FBA6 ہ or FEE9 ہ
ہ		FEEA
		FEECبہا (the clover form in the middle)

Table 6

Because of the font issues, the metadata of the Urdu text in the corpus contains the name of the Unicode-based font in which the text is stored. Any text processor or computer language that uses the data will have to normalize the usages of heh and its variations. To view the Urdu text properly in its surface form, the font in which the data was harvested will have to be applied. Differences in font mappings are not much of a problem when handling English and other Roman-based orthographies, especially when using the Unicode character set, so special attention has to be paid to the different ways fonts display surface forms of Urdu letters.

Urdu corpus construction was a significant major intellectual challenge to accomplish this research. A portion of the corpus is freely available for academic research with the copyright.

4.3.1.3 Summary

Becker-Riaz (2002) Urdu corpus is a necessary enabling resource to do Search and NER for Urdu. The corpus construction for an MRL like has many challenges that include data acquisition in a legal manner, transformation of content with varying source formats, representation of data in a consistent Unicode format with correct font representation. The Becker-Riaz corpus is an industry standard corpus that is used in many research institutions for automatic language processing.

4.3.2 Stop Words

This section describes the task of automatically generating a stop word list for Urdu to be used in information retrieval tasks. The experiment was done using monolingual English and Urdu corpora. The English documents were used to show that an algorithm for generating stop words in one language will not necessarily work in another language. Instead of creating a static list for use on all Urdu corpora, we experimented with methods that will compile a stop word list tailored to a given corpus automatically. We used two different techniques to extract stop words from two Urdu corpora. The first method was pairwise intersection and unions of the documents which showed results that were high on recall but low on precision. The other approach used a variant of uniform distribution based on word frequency. This approach showed very promising results of almost perfect recall and precision. We used the Becker-Riaz Urdu Corpus and the EMILLE (Enabling Minority Language Engineering) Urdu Corpus. The details of this work is presented in (Riaz 2007)

4.3.2.1 Introduction

A stop word is sometimes referred to as non-functional words in Information Retrieval. The purpose of identifying stop words is to intentionally not index these words to improve precision and recall. In English, stop words are words like *to, the, on, in,* etc. Since these words are found in almost all documents, they should be removed from an English search engine so there should not be a hit on these words and therefore prevent non-relevant documents from becoming part of the result.

A stop word list for English information retrieval was compiled by Fox (1992), but because Urdu is a less-studied language, some basic tools, like stop words and stemming techniques, are yet to be compiled. In this section, the process of automatically generating a stop word list for Urdu is described. The stop words found are used in Urdu search engine based on the Becker-Riaz Urdu Corpus and possibly some other Urdu news corpus if one arises. The stop words found are also used to experiment a non-linguistic approach to replace the case markers for Urdu NER.

Removing stop words for indexing purposes is one of the basic tasks of information retrieval. Stop words are removed from a corpus before an inverted index is created. Some stop words in English are *a, an, as, at, by, he, his, me, or,* and *us.* Most information retrieval systems process an existing static list against the corpus to get rid of stop words. This research presents two techniques to generate stopwords lists for Urdu. The first approach uses an iterative pairwise intersection and unions of the documents in both Becker-Riaz and EMILLE Urdu corpora. The second approach utilizes the uniform distribution of words in both the corpora. Both the approaches are general and could be applied to any genre. The Becker-Riaz corpus and the EMILLE corpus are constructed quite differently and represent different genres.

4.3.2.2 Related Work

A number of approaches have been explored to create stopword lists for English and non-English information retrieval systems. Almost all English document retrieval systems use the static stop word list suggested by Fox (1992). The Fox list is based on corpus frequencies and uses the Brown Corpus. Savoy (1999) followed the guidelines suggested by Fox to create a stop word list for French. The Savoy study entailed sorting all word forms appearing in the French corpus according to their frequency of occurrence and extracting the top 200 most frequently occurring words. All numbers were removed from the list and all nouns and adjectives more or less related to the main themes of the underlying collection were also manually removed. The Savoy study encourages using the stop word list compiled by this method on general French corpora. Savoy has applied the Fox guidelines to several European languages, but not to any South Asian languages or languages belonging to the Indo-Iranian family like Urdu, Hindi, or Farsi.

Moulinier (2004) used collection statistics and query log statistics to generate stopword lists for Korean, Chinese, and Japanese using English as a pivot language. The results showed that each approach generated different results. The query statistics were dependent upon the length of the query—that is, long queries benefited more from the stop word list and short queries did not benefit much from the existence of a stop word

list. Moulinier found that there was no statistical difference between the stop word list generated by the collection statistics and the one generated by query log statistics. This approach used WESTLAW, a commercial search engine, and the study had access to a huge pool of query logs and millions of documents for the pivot language. Lo et al. (2005) have published work for automatically building a stop word list using classical information retrieval methods and the Kullback-Leibler divergence measure. The study first formed a baseline using Zipf's law that was based on a term-rank frequency measure. Then they used a term-based random sampling approach to determine how informative a term was. The more informative the term, the more important it was. The Kullback-Leibler divergence measure determined the importance of the term. This was tested only in English. There is no published literature for a description of an approach for compiling a stop word list for Hindi or Urdu.

4.3.2.3 Automatic Generation of Urdu Stop Words

Since Urdu is an MRL and has case markers, compounding, derivation and other complex morphological properties simply using post positions and connectors could not be used as stop words. For example, the word پر /pər/ *(on)* also means پر /pər/*(wings)*. Therefore, the objective of this research is to build a stop word list for Urdu automatically. Stop words are re-defined as words which are a function of other words in the corpus. Stop words are those words that if queried will return a large number of documents, possibly the whole corpus. If too many documents are returned, then no information retrieval is accomplished. In other words, the keywords which are chosen for building the inverted index should discriminate between documents by not occurring too often, or too seldom. This is called *resolving power* in information retrieval literature (Belew 2000).

Simulating set intersection across sample English documents workable stop words are generated for English corpus. The same algorithm was tried on a subset of the Becker-Riaz Urdu corpus and the EMILLE Urdu corpus. The Becker-Riaz set of documents did not yield a single stopword (i.e., not a single word is common to all documents). The set of documents from the EMILLE corpus returned only the word کی *(kee)* (feminine

possessive). When a list was created by doing the union of intersections of all the documents in each corpus a number of content words were returned that are prevalent in both corpora besides stop words. A TF-IDF-based strategy to identify the stop words yielded acceptable stop words that are distributed uniformly across the corpora.

4.3.2.4 Data and the Gold Standard

Becker-Riaz Urdu Corpus and EMILLE Urdu monolingual written Corpus for data are used to create the gold data. The Becker-Riaz corpus consists strictly of BBC Urdu news articles published from 1999 through 2007. The EMILLE Corpus contains documents that discuss topics like commerce in India, musical instruments, homeopathy, and drama literature. We used two datasets from the Becker-Riaz corpus, the smaller Urdu dataset comprised of 200 documents averaging 336 words per document. The larger Urdu dataset contained 2500 similar-sized documents whose dates ranged from 1999 to 2004. We used 101 of the 368 EMILLE documents. The documents were preprocessed to generate a list of unique words and their frequencies for each document. A snippet of a pre-processed document compiled from the Becker-Riaz Urdu corpus is shown in below. The document is initiated by the word *Doc,* and *endDoc* signifies the end of a document. The digit with the word signifies the number of times the word appeared in the document.

<Doc 1>

6 :انکے /unke/ (*theirs)*

4 :ہوئیں /huẽ/ *(to happen)*

10 :کہیں /kahẽ/ *(where)*

50 :ممالک /mumalık/ *(countries)*

3 :خارجہ /χarʤə/ *(foreign)*

3 :قیادت /qjadət/ *(leadership)*

8 :چاہتے /ʧahte/ *(to want)*

<endDoc>

English data composed of 200 documents that are taken from bbc.com. These documents were transformed to match the structure of the Urdu documents (word: frequency).

Two types of the gold standard are used for the stop words. One was compiled with the help of three native speakers. Each document per experiment was evaluated by three different native speakers who listed the stop words that they thought did not bear any theme or content value. Stop words differences among the native speakers were resolved by the voting mechanism. A native speaker trained as a linguist was the final judge. The other gold standard was created by a rudimentary Urdu search engine created as part of this research. For each of the words returned as the stop word it was given to the search engine and it returned the number of document that matched it. If the number of documents matched were more than 85% of the total documents, then it is assumed that the word is a stop word.

4.3.2.5 Set-based Stop Word List Generation

The first technique used to find stop words was to find the set of words in the Urdu documents that are common across all the documents. We used 200 English and 200 Urdu documents and then larger Urdu data samples as mentioned in the data sections above (2500 and 101 documents). The English documents were used to show that an algorithm for generating stop words in one language will not necessarily work in another language. In the following section, we describe the algorithm for the set-theoretic approach.

4.3.2.5.1 Algorithm and Issues for Set-based Approach

First, a basic set-theoretic algorithm was chosen to find the set of words common to all the documents. Effectively, an intersection of all the words across all the documents is taken. The sample sizes were 200 and 2500 for the Becker-Riaz corpus and 101 for the EMILLE corpus.

Interestingly, when the above intersection was implemented to the resultant set returned empty for the Becker-Riaz corpus, and only one word was present for the EMILLE corpus. That means almost no one word is common enough to occur in every single Urdu document. This was an unexpected result. When the same algorithm was run on the 200-document English corpus, the resulting set contained stop words like *the, of, an, a,* and *to*. This result shows that function words in Urdu and English do not act in the same

manner and that the Urdu stop words are not common to all documents. We hypothesize many reasons for such behavior that are discussed in the analysis section. One grammatical reason could be that Urdu lacks a definite article and its indefinite article, *ek*, is more restricted in use than and an in English.

In light of the above result, the algorithm was modified for the Urdu data. Instead of calculating the intersection of all the documents, the intersections resulting from the pairing of documents were found—the final result is the union of those intersections of the pairs. The pairs were selected in sequence and randomly— the order of the pairs showed no difference in the results. The boundary conditions were handled in case there was an odd number of documents. The total number of sets was roughly half the total number of documents. A set union of all the sets returned from the first pairing was then taken. The union resulted in 442 words when run on the 200-document Urdu collection. Evaluation from native speakers determined that only 120 of those words were stop words and the rest were content words. The algorithm yielded a high recall of 98% and a very low precision of 27%. Therefore, some mechanism was needed to filter out the content words. Instead of attempting another set-based scheme, the collection of resultant sets was assumed to be pseudo-documents and was used to calculate the union of intersections again to get finer intersections of the pseudo-documents. The result of the above-nested union of intersections returned 122 words out of which 115 (average) were judged to be stop words. The precision was improved from 27% to 94%. The results were consistent for three different runs of different 200-document sets. The same algorithm was then run on the 2500-document collection of Urdu. The 2500-document collection set returned 420 words out of which 376 (average) were judged to be stop words which shows a precision of 89.5%. It is very important to note that the precision numbers listed are calculated by using the gold standard which uses the search engine instead of the words produced by the evaluators. When this list was compared with the judgment of the native speakers, the precision was reduced to 48%.

In order to compare this approach with the classic approaches of Fox for English and Savoy for French, their experiment was with Urdu data on the 2500-document collection. The most frequently occurring Urdu words were, in fact, stop words as judged by native speakers. After the first 58 words, content words started to occur periodically. Some of the content words that occurred were police, *siasat* (politics), Pakistan, and *halak* (killed). As the list progressed, the occurrence of content words become more frequent. Unexpectedly, stop words like *aisa* (such), *par* (on), and *waloon* (theirs) clustered again on the list between the top 200 and 250 most frequent words. A number of high-frequency words were the names of people and places or adjectives associated with them (e.g., *amreekee* (American). We concluded that above approach would produce a good stop word list after some manual intervention. We also concluded that the counting the top most frequently occurring words with manual intervention is equivalent to the set-theoretic approach. The results show that MRL and non-MRL languages the word distribution is quite different at least in the two examples studied.

4.3.2.5.2 Evaluation of the Stop Word List for the Set-Based Approach

The words in the three final Urdu lists that were judged to be content words by three native speakers could be considered stop words for the Becker-Riaz corpus only. They could be considered stop words for this corpus when the definition of *stopword* is that the word fails to distinguish one document from another. Those content words are so common among the documents of this corpus that a Boolean query to the corpus with one of those words could return up to 85% of the documents. From the purist point of view, the result list contains a number of content words along with the stop word list. It is acknowledged that such an interpretation may be problematic for some, but point out that the content words returned are very frequent in the corpus and are distributed somewhat uniformly. Minimally, these content words tell us about the theme of the corpus. We will show this by examining the content words in more detail.

When one of the 200-document sets was run, the seven words on the stop word list that were judged to be content words instead were *Pakistan, Hindustan, Afghanistan,*

America, soldier, Daniel Pearl, and *Omer Sheikh.*[15] Since the corpus consists entirely of BBC news articles which pointedly cover topics important to Pakistan, it is expected that *Pakistan, Hindustan, Afghanistan, America,* and *soldier* would be common to the corpus. It was not expected that the two persons' names would be common to the whole corpus, and in fact they are not. Those names were found because the number of corpus documents was so small and contained relatively chronologically consecutive news articles from the time when Omer Sheikh was prosecuted for the murder of Daniel Pearl. But at the same time these terms could be used to figure out trending in the corpus.

The names of countries and people are fairly important to information retrieval though and should not be on a stop word list that is going to be used in different genre. Named entities would be better placed on an authority list so they can be recognized as such at the time of indexing.

A careful analysis of the word list generated from the 2500-document collection shows that the words judged to be content words instead can be divided into two groups. One group can be classified as named entities (i.e., the names of people, countries, and cities). Examples of these words were *Lahore, Musharraf, Bharat,* and *Hindustan*. The second group is comprised of words that are used frequently used in journalistic articles because of the nature of journalism. For example, the term *tazjia (analysis)* appears a lot in the corpus. The titles of important world figures like *wazir-e-kharjah (foreign minister), general, sadr (president),* and *sarbarah (leader)* also appeared. The 2500-document collection contained 44 content words comprised of 17 named entities and 27 journalistic content words. As mentioned earlier, if an authority list is maintained for excluding named entities from the stop word list, then the precision of the 2500-document collection improves to 93.5%. The analysis of this result aided us to understand and mine the patterns that could be used for named entity recognition in Urdu.

---

[15] The two persons' names, *Daniel Pearl* and *Omer Sheikh,* were actually found as the four words, *Daniel, Pearl, Omer, and Sheikh.*

Given the results from the two different-sized collection sets of Urdu data, it is seen that a useful stop word list tailored to a given Urdu corpus can easily be constructed by using some manual effort.

4.3.2.6 Threshold-based Uniform Distribution Stop Word List Generation

Although the set-based approach produced the stop words that can be used after the manual intervention, the definition of stop words for set based approach can be viewed as too liberal for some. The results are too low on precision to be useful given the usual definition of stop words. To improve the precision of results, a two-pronged approach is pursued with the threshold-based strategy.

First, the definition of the stop word was altered for this experiment to be more conservative regarding which words are considered to be stop words. Given a document, only those words are considered stop words that do not give any indication of the topic, theme or content of the document. This definition is aligned with the manual construction of gold standard by native speakers. Secondly, find words which are evenly distributed across the corpus. The word frequencies are used to determine if a word is evenly distributed within the document and also across other documents.

This approach requires an evaluation of the frequency of each word in a document across all other documents in the corpus within some slack window. This is a very computationally expensive approach. Consider if we have *n* documents and each document has *m* words in it, the then the computational complexity is asymptotic in terms of *m*. Given that Becker-Riaz corpus has about 6 million words and EMILLE corpus has roughly about 7 million words, implementation of such an algorithm is prohibitive. To solve computational tractability problem, a threshold based approach is attempted whose goal is to reduce the number of words in the document that are considered stop word list candidates.

4.3.2.6.1 Algorithm & Issues for the Threshold-based Uniform Distribution

The algorithm was tried on 2500 documents from the Becker-Riaz corpus and 101 documents from the EMILLE corpus. Each document in the respective corpus is processed

to get the highest frequently occurring word. All the words in a document that are above a certain threshold say x, are extracted as stop word candidates. For example, all the words that occur in a document with more than 60% of the highest frequency word are extracted as potential candidates for stop words. A number of thresholds were tried ranging from 10% to 70%. As expected, if a threshold of 10% is used the recall is very high but the precision is quite low. Similarly, if we use a high threshold value, then the recall is low and the precision is high.

After experimenting with many different values, 55% threshold is considered optimal for Becker-Riaz corpus and a threshold of 40% for the EMILLE Corpus. These thresholds still yielded some content words like *(American)* امریکی in the results of the Becker-Riaz corpus. The EMILLE results showed almost all the candidates were actual stop words. The results are discussed in the next section. To further refine the algorithm process, Inverse Document Frequency (IDF) of the words is used in the new reduced candidate word set. An inverted index was created for the reduced candidate file and IDF was calculated for each word. IDF proved to be a useful measure because it assigns the importance to the word according to its rarity in the corpus. If there are *N* documents in the corpus and term t_i occurs in n_i of the documents, then the Inverse Document Frequency (IDF) for the term t_i is:

$$IDF(t_i) = \log\left(\frac{N}{n_i}\right)$$

Equation 13

In the modified algorithm, the IDF value for a term is normalized to be a value between 0 and 1 using the following formula:

$$IDF(t_i) = \frac{\log\left(\frac{N}{n_i}\right)}{\log(N)}$$

Equation 14

After examining the list, a cut-off value is used to determine the list of stop words. The cut-off value is determined by the threshold used to produce the contents. The smaller the threshold *x,* the smaller the cut-off value for IDF. For the Becker-Riaz corpus, when the threshold is 10%, the cut-off is 0.2, but when the threshold is 50%, the cut-off value is 0.4. For the EMILLE corpus, the cut-off is 0.75 when the threshold is 10% and 0.84 when the threshold is 40%.

4.3.2.6.2 Evaluation of Stop Word List for Threshold-based Uniform Distribution

In the algorithm given above, there are two variables. First is the number of documents and the second is the value of the threshold x. The results in this section are based on the different values of the threshold parameter when the number of documents is kept constant. The algorithm was run on the set of 2500 documents from the Becker-Riaz corpus and 101 documents from the EMILLE corpus. The algorithm is designed in such a way that words in the documents that were returned when the threshold was 60% will also be returned when the threshold is 40%. Therefore, one can measure the recall of the algorithm at a certain threshold by examining the candidates at the lower thresholds.

Experiments results were evaluated by native speaker's judgments of a stop word. The evaluation of Becker-Riaz corpus with thresholds ranging from 55% to 75% resulted in the precision decreasing when the threshold was taken more than 60%. A cursory evaluation showed the recall was low for 60% threshold. For example, for a document the word کا *(ka – male possessive*) had lower frequency count than the word *computer* and therefore, کا did not show as a candidate for the stop list. A detailed inspection of the returned results showed that if a particular stop word was missed from a document it was soon identified as stop word in another document. We found that threshold of 55% produced all the stop words in the documents. When the threshold was 55%, the list also had a bunch of content words; therefore, the precision was low.

After reaching the optimal threshold for the Becker-Riaz corpus, we used the same approach on the EMILLE corpus. The results showed that at 60%, almost all the

documents' candidate words were stop words. Threshold of 10% is used see the recall for different thresholds.

The results are show in a tabular form:

Threshold	40%	50%	60%
Recall	95%	77%	66%
Precision	50%	76%	84%

Table 7

Both the Becker-Riaz corpus and the EMILLE corpus have a few content words that ranked consistently as stop words. For example, the word امریکی *(American)*, and the word عراق *(Iraq)* were listed as stop words. For the EMILLE corpus, the content words were کتب *(Arabic for book or fist part of the phrase 'book house')* and the word زبان *(language or tongue).* The IDF measure and the cut-off values increase the precision signifcantly. The precision results for the threshold of 55% for the Becker-Riaz corpus and the 40% threshold of are shown below.

	IDF cut-off	Precision
Becker-Riaz	0.5	87%
EMILLE	0.84	91%

Table 8

By looking at the words in the EMILLE corpus, it can be seen that precision would be much higher and closer to 98% if there were no typographical errors in the corpus.

Note: As in the set-based approach, the evaluation was done by three native speakers of Urdu.

4.3.2.7 Discussion

This section explains why a mechanism for stop word generation is needed for a MRL like Urdu and the current algorithms to generate stop words for English and other European

languages cannot be used. We used English as an example language to elaborate that claim. Some of the reasons are discussed below:

- Urdu orthography and morphology permits merging of some of those words together to form a new word. For example, the English word *his* can be written as اسکا or اس کا, where the first occurrence is the combined version of the second. Each of these words is a stop word. If the words are combined to form another word, they become a separate token and will not be picked up as a stop word. This shows combining of words in Urdu morphology applies to stop words as well as compound nouns like *Bawarchi Khana (kitchen).*
- Urdu has gender marking on stop words. Therefore, two different tokens are generated depending if the object or subject is male of female entity. An example is کا *(ka)* and کي *(ki).*
- Urdu lacks a definite article and its indefinite article, *ek,* is more restricted in use than *a* and *an* in English.
- Urdu equivalent of the English term *that* is much more restricted in use in Urdu as compared to English.

    - Written Urdu is prone to orthography errors which we saw that during the evaluation of results. In the EMILLE corpus, such issues are prevalent. These errors are numerous enough that they skew the results. The transcribers or people who keyed the text consistently confuse one of the vowels in Urdu.

Given the issues above, the methods to generate the list for English are ill-suited for Urdu. The static list of stop words that is generated by one corpus may not be useful for other corpus. For example, if an author favors combining words, then the stop words that form the combined words will not be useful. Becker-Riaz corpus is constructed from the news filings from various authors hence there are many different formats to represent stop words. Also, the stop list from EMILLE corpus will not be useful to the Becker-Riaz corpus because misspelled stop words in EMILLE are considered content words, and therefore reduce recall for Becker-Riaz corpus that has fewer errors.

EMILLE corpus has higher precision because of the large document size and the diversity of topics across chapter and within a document. The Becker-Riaz corpus is news-based and focuses on South Asian events and has few data points that do not vary across time. For example, words like *Musharraf*, *Pakistan*, *India*, *cricket*, and *America* occur with higher frequency in the corpus, the word وہ *(that)* consistently occurs less than the content words as *America, Iraq, etc.*

4.3.2.8 Summary

This research shows an alternative approach to finding stop words for Urdu text in a given corpus. The approach presented here from previous approaches reported in either MRL or non MRL which require taking the most frequent words in a corpus. The stop word approach contributed in this research is general and robust to produce stop words for any corpus in a particular domain. The results showed the list generated from the given approach is very useful for creating inverted indices for our IR system which queries the Becker-Riaz Urdu Corpus or EMILLE corpus. The set-based approach also found a number of named entities that helped to determine some patterns to determine named entities. This approach was used to enrich some of the entries in a small look up authority file in Urdu NER development. The threshold-based uniform distribution approach paired with IDF provides good results to extract a stop word list given a corpus

4.3.2.9 (Threshold-based – Becker-Riaz Corpus – 55%)

کیا

/kja/*(what)*

کر /kər/ *(to do)*	ہیں /hẽ/ *(are)*	اور /or/ *(and)*
سی /si/ *(modifier)*	پر /pər/ *(on)*	ہے /he/ *(is)*
عراق /erak/ *(Iraq)*	اس /ɪs/ *(this)*	میں /mẽ/ *(in)*
یہ /je/ *(this)*	کو /ko/ *(to)*	کے /ke/ *(possesive)*
ان /ɪn/ *(their)*	کہ /ke/ *(that)*	کی/ki/*possesive*
امریکی/əmrɪki/*american*	کا /ka/ *possesive*	
ہی /bi/	نے /n̪e/ *(form of be)*	
ایک /ek/ *(one)*	سے /se/ *(modifer)*	

4.3.2.10 (Threshold-based – EMILLE Corpus –40%)

ان/un/ / *(their)*

ےا /ja/ *(or)*

کتب /kutəb/ *(book)*

کےا /kja/ *possesive*

و *(urdu letter)*

تھا /t̪ʰa/ (*was)*

نہےں /n̪əhĩ/ *(no)*

ہے/je/ *(this)*

پر /pər/ *(on)*

زبان /zəban/ *(tongue)*

کہ /ke/ *(that)*

کر /kər/ *(to do)*

نے /n̪e/ *(form of be)*

ہےں /hẽ/ *(are)*

کو /ko/ / *(to)*

اور /or/ *(and)*

مےں /mẽ/ *(in)*

کے /ke/ *(possesive)*

کا /ka/ *(possesive)*

اس /us/ *(that)*

ہے /he/ *(is)*

سے /se/ *(modifer)*

کی/ki/*possesive*

4.3.3 Stemmer

In this section we describe the construction of a stemmer that will be required for Urdu search. Currently, we know of no Urdu stemmer for Information Retrieval that will improve recall of a system. There are morphological analyzers for linguistic processing but they are not fit to improve the recall of a system. Details of creation of Urdu stemmer are available in (Riaz 2007).

4.3.3.1 Introduction

A *stem* in linguistics is the combination of the basic form of a word (called the root) plus any derivational morphemes, but excluding inflectional elements. Some researches make a distinction between the concepts of *stem* and *root* but the difference has no practical value for stemming purposes. Alternatively, a stem is the form of the word to which inflectional morphemes can be added (Porter 2001).

A *morpheme* is the smallest language unit that carries a semantic interpretation. Generally, morphemes are a distinctive collocation of phonemes (the free form *pin* or the bound form -s of *pins*) having no smaller meaningful members. *Inflection* refers to modification of a word. More precisely, a word is modified so that it reflects grammatical information such as word gender, tense, or person. The root of the English verb form *destabilized* is stabil- (alternate form of *stable*); the stem is de·stabil·ize, which includes the derivational affixes de- and -ize, but not the inflectional past tense suffix -(ed).

In languages with very little inflection such as English and Mandarin Chinese, the stem is usually not distinct from the “normal” form of the word. However, in other languages like MRL, stems are more noticeable. For example, the English verb stem *eat* is indistinguishable from its present tense (except in the third person singular).

4.3.3.2 Stemmer

A stemmer is a computer algorithm to reduce the words to their stem, base, or root form. A stemming algorithm has been a long-standing problem in Information Retrieval. The process of stemming, often called *conflation*, is useful in search engines, natural language

processing, and other word processing problems. For example, a stemming algorithm reduces the words *fishing, fished, fish,* and *fisher* to the root word *fish*.

Stemming is mostly used to increase the efficiency of a search engine—more specifically, to increase recall to a large degree. Precision is sometimes increased or decreased depending upon the information need of the user. Earlier stemmers were rule-based written in BNF notation. The first paper on the subject was published in 1968 by Julie Lovins who listed about 260 rules for stemming the English language. She used *Iterative Longest Match heuristic,* which means that a rule is preferred over the other rule when its right side matches the most characters. The most notable work is presented by Martin Porter in 1980 who simplified the rules of Lovin to about 60 rules. The algorithm attributed to this work is called Porter Stemmer and is most widely used in search engines.

Stemming can be classified as weak stemming and strong stemming. Weak stemming entails executing a few selected rules (e.g., one can just decide to use the rules that deal with plurals in English). The use of rule-based stemmers is very helpful because upon encountering new words, one can change the grammar instead of trying to solve the problem by ad hoc approaches (Belew 2000). Stemmers are common elements in query systems because a user who runs a query on *cars* probably also cares about documents that contain the word *car* (without the s).

4.3.3.3 Stemming Problems

The fundamental problem with any stemming technique is that the morphological features being stripped away may well obscure differences in the word's meaning. For example, the word *gravity* has two word-senses, one describing an attractive force between any two masses and the other having to do with a serious mood. But once the word *gravitation* has been stemmed, we have lost the information that might constrain us to the first interpretation, this means that conflation has happened (Belew 2000). Porter (2000) 2describes the problem of under-stemming, over stemming and mis-stemming based on suffix stripping algorithm known as the Porter Stemmer (Porter 1980). Porter (2000) mentions three classes of suffixes:

Attached: a particle word attached to another word.

Inflected: part of the grammar to indicate the tenses and other aspects of the grammar e.g. telephone → telephoned.

Derivation: It forms a new word often with a different grammatical category or a different word sense e.g. clean → cleanliness.

Under-stemming happens when the stemming algorithms leaves the suffix attached to the word. Over-stemming happens when the algorithm removes too much of the suffix. Mis-stemming is removing suffix which was actually part of the word. Porter (2000) suggests that over-stemming and mis-stemming can be corrected by the use of dictionaries.

4.3.3.4 Language Dependency

English stemmers are fairly trivial with only occasional problems in the retrieval performance such as *axes* being the plural of *ax* as well as *axis.* However, stemmers become harder to design as the morphology, orthography, and character encoding of the target language becomes more complex, mostly it is the inflection of the language. For example, an Italian stemmer is more complex than an English one (because of more possible verb inflections), a Russian one is more complex than Italian, an Arabic stemmer is quite more than all of the above because of orthography, and high inflection. For example, for plurals, Arabic has a word for singular, two-count plural, more than two-count plural. Since Urdu is an MRL, and its morphology is a combination of its influencing languages, the hypothesis it will be at least as more complex as the most morphology complex language.

4.3.3.5 Urdu: A Challenging Language for Stemming

Urdu vocabulary is a composition of many languages and adopts words from other languages with ease. Besides having its own morphology, Urdu morphology is strongly influenced by Farsi (Persian), Arabic, and Turkish. Therefore, Urdu vocabulary is composed of the above-mentioned languages along with many Sanskrit-based and

English words. The morphological rules of each word follow the morphology of the language where it is borrowed. For example, the word *pachim* (Hindi) and *maghrib* (Arabic) both mean the direction *(west)* in English and are both Urdu words as well. The rules of stemming of *pachim* need to use the Sanskrit or Hindi rules and for *maghrib* Arabic rules will need to be used.

Urdu is a bi-directional language with an Arabic-based orthography. Bi-directional means that it is very common in Urdu to see an English word written in Latin-based characters. Sometimes an English word is written phonetically with Urdu characters. For example, *(pub)* is written as پب . Although Urdu has Arabic orthography, its grammar is based on Sanskrit and Persian. Urdu has gender marking on its parts of speech. For example, *paharh (mountain)* and *paharhi (hill)* and a stemmer will conflate both word to *pahar*. Therefore, stemming Urdu words will increase recall and also conserve space usage of the indices.

4.3.3.6 Approaches for Stemmer for Urdu

There are two ways to build a stemmer, statistical-based and rule-based (linguistic-based). Rule-based stemmers use prior linguistic knowledge or the morphology of the language to form stemming rules. In contrast, the statistics-based approach uses statistical principles to infer the word formation rules (morphology) by analyzing the corpus. One general statistical stemming approach uses Bayes theorem to figure out the probabilities of the suffix given the word and then uses popular link based Hits algorithm to assign scores to the stems (Goldsmith et al. 2001). In this paper, only the rule-based approach is explored.

4.3.3.7 Related Work

Although there is no published work in the IR community describing challenges in Urdu stemming, there has been considerable work done towards computational morphological analysis of Urdu. Butt (2001) and Rizvi (2005) describe the computational analysis of different parts of speech in Urdu. Their work is very important for computational Urdu processing but is focused towards theoretical, computational linguistics. Their work certainly can be used to build the rules for an Urdu stemmer. A morphological analyzer is

available at Computing Research Laboratories (CRL). After a brief analysis of the results produced by CRL morphological analyzer and examining the root forms, a number of false positives and errors were observed. We theorize that this is because CRL software used the same rules for Urdu as Arabic and Farsi. Savoy (1993) uses a dictionary-based approach to stem French words in the corpus. This work is quite interesting, but unfortunately, there are no machine-readable Urdu dictionaries available to see the utility of this approach to Urdu.

There has been significant work done on Arabic stemmers. Most of it is statistical and heuristics-based (Chen et al. 2002). Although Farsi and Arabic are written with *similar* (not the same) scripts, stemmers of those languages are not adequate for Urdu stemming. Arabic stemmers produce a large number of over-stemming and mis-stemming errors for Urdu because of its high inflection and complex grammar. Farsi stemmers produce a number of incorrect stems because it accurately stems only Farsi loan words and errors on native Urdu and Arabic loan words etc.

4.3.3.8 Heuristics based on Urdu Parts of speech and Morphology

This section also explores avenues of building an Urdu stemming from parts of speech tags and other heuristics. For each category, discussion points are raised that were considered during the research.

a. Nouns

Following are stemming issues related to nouns

- Most of the issues regarding nouns pertain to gender marking—determining if the noun is masculine or feminine: marked nouns for gender, e.g., lardka (boy) and unmarked nouns, e.g., **جہازي** jahazi (sailor)
- Arabic loan words that end with "-at" e.g., **قیمت** *qeemat (price*)
- Persian loan words
- Indigenous words

- Noun Plurals

b. Pronouns

- Is the heuristic correct that to not process pronouns because their length is not more than four?

c. Adjectives

- Change in marked adjectives for agreement, for gender (masculine/ feminine), number (singular/plural), and case marking

d. Postpositions

- Is the heuristic correct that to not process postpositions because their length is less than four?

e. Verbs

- Stem according to different forms of verbs: root form, imperfect participle, and perfective participle

f. Exception List

- Recognize number, dates, months. If one can do that it aids in NER tasks for TIMEX and NUMEX

g. Persian element in Urdu

- use of izafat (possessive marker)
- word-forming affixes

h. Arabic elements in Urdu

- Trilateral root. It is one the basic structures of the Arabic root.
- Arabic definite article *al*

i. Use of Dictionaries

- Porter (2001) claims that over-stemming and mis-stemming errors can be corrected by using an online dictionary. There is no machine-readable Urdu dictionary available to the community. Therefore, another method needs to be devised to remedy these errors.

4.3.3.9 Weak Urdu Stemmer

A prototype of an Urdu stemmer is implemented that incorporates four rules for processing plurals and possessives and a heuristic for skipping a word, so it does not get stemmed. It was observed that the order in which the rules are executed is important, and changing the order improved the results considerably. The prototype is implemented in Java on Windows machine. The quality of the stemmer is considered adequate to do research in Urdu, but for a more robust actionable work, a more robust stemmer needs to be created for Information Retrieval purposes.

The possessive rule is a gender-marking rule for feminine marking—it addresses both the noun marking and the adjective feminine marking. The plural rules do not completely encapsulate all the morphological variations. The plural rules are for indigenous Urdu words, and there is one rule for plural in Persian. One very interesting phenomena was observed while implementing the plural rule. Some root forms of Urdu are shared amongst the indigenous form, Persian form, and Arabic form, and the word has to be processed by each of the rules (not one of the rules) to reach the proper root form. The side effect of this process is that recall is increased, but precision is decreased considerably because polysemy is introduced. The following example can illustrate the phenomenon.

If we derive the root form of the Urdu word *khabr* **خبر** *(news)*, then it could arrive from many different source words with different meanings. The possible non-stemmed word could be a loan word from another language or indigenous to Urdu. An example follows:

اخبار (a*khbar*): newspaper in Urdu and Persian, but plural for news in Arabic

خبر (*khabr*): news in Urdu, Arabic, and Persian – root form

خبریں *(khabrain)*: plural of news in indigenous Urdu and Hindi

اخبارات *(Akhbarat)*: plural of newspaper in Urdu and Persian

The results of the prototype are encouraging but far from being complete to be an actionable stemmer. The stemmer was run on 1065 words taken randomly from Becker-Riaz corpus, and the four rules were applied to each word only once. Out of 1065 words, 569 were treated by the non-stemming heuristic and four rules for stemming. Out of 569 words, 211 were stemmed, and, out of 211 words stemmed, 32 were stemmed incorrectly. The reason for the false hits was unimplemented rules—words that matched the Persian profile but were adjectives in Arabic. Additionally, some transliterated words of English like **ائرمارشل** *(Air Marshall)* and **اسکواڈرن** *(squadron)* were found in the list. The algorithm stemmed squadron **سکوڈ** *(squad)* correctly but marked Air Marshall as **ئرمر** aim because it mistakenly assumed that it was the Persian plural.

4.3.3.10 Stemming Evaluation

There are many ways to evaluate a stemming algorithm. One of the most basic methods is called direct assessment method in which performance of the stemmer is judged by examining its behavior when the algorithm is applied to a sample word. The utility of this method in isolation is not quite clear. It is typically used with other complementary techniques. The direct assessment method was used to analyze the Urdu stemming. There are two other popular techniques for the evaluation of stemming algorithms. One is based on the discipline of Information Retrieval where recall and precision measures are used to judge the performance; an increase in recall indicates a good stemmer. Error counting approach for the evaluation of stemming algorithm is quite robust and desirable along with the direct assessment method. Error counting scheme counts the over-stemming and under-stemming errors by using a sample of grouped words.

4.3.3.11 Summary

Urdu – a MRL – brings in a myriad of challenges for stemming. The prototype was written to motivate the challenges associated with the stemming of MRL Arabic Script languages. Evaluation of stemmers for morphologically rich languages like Arabic (Darwish 2002) and

the challenges faced during the creation of weak stemmer for Urdu shows that stemmer creation of Urdu an MRL will be a significant task. Darwish's doctoral work was to create a light stemmer for Arabic (Darwish 2002). In order not be distracted from the main research goal, creation of a complete, robust stemmer is postponed until this research is complete. The current stemmer is adequate to show the utility of stemming in Search and lack thereof in (NER) shown in section 4.5 and in chapter 5. Initial results show that the rudimentary stemmer can show the increase in recall for target queries for Search in Urdu and hurts the performance of NER in MRL languages.

4.3.4 Baseline and Evaluation for Urdu Search

The goal of conferences like TREC, TIPSTER, NTCIR, CLEF is to judge the performance of different algorithms. Most of these conferences have tracks that deal with new and innovative information retrieval problems, but none has tackled to work with Urdu data, primarily because of the lack of resources. One of the contributions of this research is to create a baseline for Urdu IR evaluation along with resources necessary to do the task. In this contribution, we explore that strategy for the creation of the test reference collection for Urdu Information Retrieval. Details are available in (Riaz 2008).

4.3.4.1 Introduction

Retrieval Performance evaluation is based on a reference test collection and an evaluation measure. The reference collection consists of a test corpus, a set of queries, and set of the relevant document for each query. Queries are constructed by the domain experts, and the relevant documents are selected by experts. Query generation experts and relevant document selectors need not be the same. A reference collection and query relevance judgments for 200 Urdu documents from the Becker-Riaz Urdu corpus is presented for gold data and to create a baseline. This is quite a challenging task for Urdu because of the dearth of specialists who understand the technicalities of the task at hand, which is creating a test reference collection and its usefulness in Information Retrieval discipline. TREC methodology is followed to create relevance judgments.

There are many measures proposed in the IR literature to judge the performance of an IR system. For some measures, the ranking of documents is baked in, while others do not use the ranking information to judge the performance. In experiments, both types of measures are used to create a baseline the Urdu IR system. Some of the widely used measures are Recall-Precision, F-measure, Mean Average Precision, and R-Precision. A baseline is presented with operability of the reference collection with Boolean searching and Ranked Retrieval (Vector Space Model) techniques to measure the Recall and Precision of the results

4.3.4.2 TREC test reference collection

The TREC collection, also called the TIPSTER collection, was initiated under the leadership of Donna Harman (Bates, 1999) at National Institute of Standards and Technology (NIST) Conference. The goal was to have the collection that consisted of millions of documents, provide uniform scoring procedures and forum for organizations who are interested in comparing their results (Bates, 1999). TREC has test reference collections for many languages but not for Urdu.

The search results of the participant systems are run through the same evaluation system so consistent evaluation results are seen and the participants can compare and contrast results with each other. The TREC conference has many different tracks, like Ad hoc, novelty, routing, etc. Like test collections, the TREC testbed consists of three essential parts: the documents in the test collection (corpus), the example queries (*topics* in the TREC domain), and a set of relevant documents for each example query. TREC document collection (corpus) is composed of various genres like news articles, computing reviews and legal documents to mention a few. All documents in the collection are tagged with SGML-like tags. There is no stemming done on the words and stop words have not been removed. Some statistics are provided for each genre, an example of statistics for a genre is given in Table 7.

Genre	Size in MB	#Docs	Median Words/Doc	Mean Word/Doc
Financial Times 1991-1994	564	210,158	316	412.7

Table 9

A skeleton of TREC document is given below:

<DOC>

<DOCNO>document_number</DOCNO>

<TEXT>

document_text

</TEXT>

</DOC>

There could be other tags present in other sub-collections, or the type of the data, e.g., new articles have the tag of <h1> to indicate the title, <author>, and <dateline>. The details for each format can be obtained from TREC's Web site (NIST 2017).

The example queries or topics in TREC nomenclature are created by experts who have long been in the information seeking business. The query (topic) is represented in an SGML-like snippet. An example topic from TREC 2006 Ad hoc track given below

<top>

<num> Number: 301

<title> International Organized Crime

<desc> Description:

Identify organizations that participate in international criminal

activity, the activity, and, if possible, collaborating organizations

and the countries involved.

<narr> Narrative:

A relevant document must as a minimum identify the organization and the type of illegal activity (e.g., Columbian cartel exporting cocaine). Vague references to international drug trade without identification of the organization(s) involved would not be relevant.

</top>

The format and the comprehensiveness of the topics have varied over various TREC conferences. Since TREC corpus contains millions of documents it is unrealistic to provide relevance judgments for each query. Therefore, a pooling method is used to judge the relevant documents. Pooling method combines the top *N* results for each participant system, an intersection of these results are shown to judges for evaluation. Pooling method has to be quite effective and empirically correct (Bates 1999).

4.3.4.3 Urdu Test Reference Collection

One of the major hurdles for Urdu processing is the lack of baseline evaluation mechanism for results. There has been some work done on other Indian language evaluation through Forum for Information Retrieval Evaluation (FIRE) (Majumder et al. 2007). This section describes the process of creating a gold standard (baseline) for Urdu that is typically available for other languages. Ad hoc track is followed to build the Urdu baseline.

Information retrieval is fundamentally a user-driven task where the search engine is trying to satisfy an information need of a user. This is evident from users reissuing queries after unsatisfactory results. In the absence of a test collection, one can always do user studies for evaluation. But this is an expensive proposition in the absence of funding. TREC methodology was used order to create the Urdu baseline. With the absence of any funding, and most importantly the lack of human resources to create the queries and relevance judgments the scale of TREC cannot replicate for creating an Urdu baseline. The baseline and relevance judgments can be created for the two available Urdu corpora, the EMILLE corpus (300 documents) and the Becker-Riaz corpus (7000 documents). EMILLE is not suited for creating relevance judgments and queries because it contains documents that are quite large in size, discuss quite disjointed topics and are quite unlike Web documents. Becker-Riaz corpus consists of Web news stories and therefore could be used for generating relevance judgments and queries.

Unfortunately, there are few researchers in the IR community who know Urdu and IR concepts and most importantly could volunteer to create topics and relevance judgments. For this research 200 documents for the test, corpus are chosen because it is manageable to create relevance judgments and create topics and supervise the work produced by topic and relevance judgment creators. Two volunteers from Pakistan and one from the United States participated to generate queries and relevance judgments. The resources from Pakistan were university students and the resource from the United States was a lifelong newspaper reader who had a good command of the politics and news in South Asia. The resource in the United States was quite naive about the use of computers and

the field of Information Retrieval, so it required quite a bit of coaching before giving the task and scrutinizing of work after the deliverable.

The documents in the test reference collections were kept in Corpus Encoding Standard (CES) (Ide, et al. 2000) a departure from TREC format. We chose this format because Becker-Riaz corpus is metadata rich and we plan on using this metadata for algorithm development for the larger goal for enhanced search. The metadata contains a number of named entities

The topics were constructed in the same fashion as TREC format, but the snippet is well-formed XML instead of the SGML-like snippet in TREC which is not well-formed XML. The title information of the topic is in Urdu, but other metadata like a description of the topic are in English. An example of topic is given below:

<topic>
<number>1</number>
<urdu.title> کرکٹ کپ </urdu.title>
<english.title>Cricket Cup</english.title>
<description> Identify articles that discuss the news stories about different cricket tournaments, opinions about the matches and results and performance of batsmen and bowlers. The documents could be about the World Cup or other cricket tournaments where a number of cricket playing countries participate for a prize </description>
<narration> A relevant document must as a minimum identify the document that relates to cricket tournaments where a major prize is awarded </narration>
</topic>

The relevance judgments are provided in a table where the set of documents that match the topic are listed. Some statistics of the 200 Urdu documents are given below:

Genre	Size MB	#Docs	Median Words/Doc	Mean Word/ Doc
Becker-Riaz	1	200	119	183

Table 10

4.3.4.4 Baseline evaluation

To check the operability of the test reference collection, evaluation was done on a home-grown Boolean retrieval engine, Apache Lucene, Lemur from UMASS, and Terrier from University of Glasgow search engines. Lemur (a C++ based engine) had encoding and XML processing issues. When a document was transformed into TREC format to circumvent its XML processing issues, Lemur indexed Urdu fine, but retrieval did not recognize the query in Urdu. The transformation of XML documents into TREC like format is not desirable because a lot of useful metadata information is lost that way from the Becker-Riaz corpus, e.g., keywords, author, date, word count, and type of document (news story, column, etc.). Lucene appeared to be the most versatile to process most Unicode, XML, and metadata-aware search engine.

The words in the corpus were not stemmed and stop words were not removed from the documents. The purpose of keeping the raw corpus was to establish a baseline and then use stemming and stop word removal techniques to measure the Recall and Precision of the same queries. The results showed that the Boolean engine works accurately with logical OR and logical AND queries in isolation, but was not working well with the combination of AND-OR queries. Lucene is a combination of Boolean and Vector Space Model and has fielded search capability. Experimentation on Lucene entailed out-of-the-box Lucene engine without any weights on terms. Three types of queries or topics were used to establish the Urdu baseline: simple queries, concept queries, and phrase queries. Simple queries are one-word queries, concept queries can be multiword and are designed to retrieve documents that may or may not contain the keyword, and phrase queries contain Urdu phrases. Some results of each query type are given below. All queries contained named entities as shown in the examples.

Simple query پرل *(Pearl)* used to search for news articles about the journalist Daniel Pearl. Initial testing showed that the fielded search on *Title* retrieved about 6 of 11 relevant documents, but all the retrieved documents were relevant. The *Para* fielded search

resulted in all the 11 documents. When we used the query اسرائیل *(Israel)* 6 out of 25 relevant documents were retrieved when we used *Title* fielded search. The results missed a number of documents because the query was not اسرائیلی *(Israeli).* These two terms will conflate to one after stemming and provide a much better recall. The results show that relevance judgments can be used to identify the reasons for low recall as shown in the query *Israel.* A concept query containedپاکستانی عدلیہ (*Pakistan's Judicial System)* to retrieve news stories about situation of Pakistan's court system. There were no relevant documents retrieved when we used *Para* fielded search. The *Title* fielded search retrieved 1 of the 4 relevant documents. This is mostly because Lucene is a keyword-based engine. For phrase queries the following terms are used نیپال میں ماؤنواز باغیوں کی سرگرمیاں *(Activities of Maoist rebels in Nepal)*. This query had 13 relevant documents but the *Title* fielded search retrieved 104 documents and the *Para* fielded search retrieved 109 documents. This is because Lucene did a logical-OR of all the query terms including the two stop words in the query. When the stop words were removed, the *Title* fielded search retrieved 4 documents 3 relevant and 1 non relevant about rebels in Fiji. The *Para* fielded search retrieved 6 documents of which 3 were relevant. The results from the phrase query suggest that the relevance judgments can be used to identify the importance of stop word removal to improve precision and recall of the IR system.

4.3.4.5 Summary

Since there is no robust test reference collection for Urdu, a small test reference collection can be used to establish a baseline for Urdu processing algorithms. Becker-Riaz corpus is best suited for the Web retrieval algorithms because it is comprehensively composed of news articles from the Web. The baseline also shows that the relevance judgments can aid in the experiments to show the utility of named entities.

4.4 Urdu and Hindi Comparison

Urdu and Hindi are considered sisterly languages. Urdu shares its grammar with Hindi but they are written in two different scripts. As part of this research, a number of peer-reviewed articles have been published over time as referenced in the bibliography. Frequently, reviewers added comments that either a linguistic resource exists in Hindi that could be used for Urdu or Hindi could be used to accomplish through transliteration because the difference between them is only the script and the languages are mutually intelligible. Most of these researchers are well-known names in the NLP community. The purpose of this comparison is to show the difference between them is much more than script as these languages are evolving. Hindi is a vibrant resource-rich language, but Hindi resources are not suitable for Urdu research in their current state despite the similarity of the two languages.

4.4.1 Introduction

Urdu and Hindi share a complex relationship. Together they boast one of the largest populace who understands them and calls either one of them as their national language. They are the languages of South Asia—Urdu is the national language of Pakistan, and Hindi is the official language of India. India does not have a national language because of the number of regional language-sensitive provinces. Urdu is one of the official languages of India. Urdu is written in the Arabic script, and Hindi is written in Devanagari script. While researching Urdu named entity recognition, we wanted to use some Hindi language resources like gazetteers, and online dictionaries to look up names of cricketers, movie actors, capitals of states, and countries. Such resources are not available for available for Urdu. We realized in early research stages that Hindi and Urdu were behaving as separate languages. Moreover, we needed to learn Devanagari script to proceed further. For a casual observer and for NLP researchers not completely familiar with both languages, Urdu and Hindi are mutually intelligible with different scripts. In this position paper, we suggest that Hindi and Urdu, although grammatically very similar, cannot be treated as

the same language for researching computational linguistics, machine learning, and information retrieval—at least with the current set of tools available for both languages. We will show that some researchers treated Urdu and Hindi interchangeably with mixed results. This is because Hindi has quite a vibrant set of enabling technologies for machine learning, computational linguistics, and information retrieval whereas research in Urdu is still in its infancy. Some examples of these enabling technologies which are available for Hindi are: online dictionaries, Wordnet, stemmers, stop word lists, gazetteers for named entity recognition systems, part of speech taggers, baseline for evaluation like Forum for Information Retrieval Evaluation (FIRE), training data for multiple machine learning tasks to name a few. Some of the mentioned tools for Urdu are available through CRULP and other resources in varying developmental stages. This section provides a brief overview of Urdu in the context of the similarity between Hindi and Urdu. It also discusses the differences between Urdu and Hindi and motivates why a position needs to be taken for making progress in Urdu computational processing. Quantitative analysis is also presented to highlight the differences between Hindi and Urdu to reinforce this position. Machine Translation, Machine Transliteration, Information Retrieval, and Named Entities Recognition areas are explored to show case the differences between Hind and Urdu.

Urdu, among all above languages mentioned, has a unique case in that it shares its grammar with Hindi. The difference is some vocabulary and writing style. Hindi is written in Devanagari script. Because of the grammatical similarity, Hindi and Urdu are considered one language with two different scripts by many linguists. It suggested that there are a growing number of dissenters of this categorization among South Asian language researchers who are familiar with both languages. Because of its rich morphology, word-borrowing characteristics from different languages, and its native capability to draw upon expressions and references from different cultures and religions, Urdu is widely considered the language of the poets as evident from the large volume of poetry and poets produced in South Asia (Ghuravi, 2013). One of the most prolific Hindi movie lyricists and songwriter Gulzar writes songs in Urdu for Hindi movies (Press Trust of India, 2015). This is similar for a scientist to be well versed in German and French in 1800 and

1900s to understand scientific communications and published research (The Atlantic, 2015).

4.4.2 Background

While researching Urdu computational processing, a claim can be made that Urdu and Hindi are the same language in two different scripts (King 2001). According to this theory and lack of computational resources for Urdu, any computational model or algorithm that works for Hindi should also work for Urdu. Some Hindi examples for resources are the baseline for evaluation methodologies that are designed for Hindi (FIRE, 2010), gazetteers for named entity recognition, and WordNet for Information Retrieval to name a few. Ahmed (2011) tried to create a knowledge resource of Urdu using transliteration from Hindi WordNet claiming that structure and vocabulary of the Hindi and Urdu and the only difference is the script. Visweswariah (2010) claimed that Urdu and Hindi share common morphology, phonology, and grammar but different script attempted to use machine translation between Urdu to English using Hindi to English Parallel Corpus. These experiment's details and some gaps in the simplification assumption of two vibrant languages are explained in detail in Section 4.6.4.

The following section describes in detail that the *one language two scripts* theory for Urdu and Hindi is invalid in many situations and specifically for computational processing. Although a lot of research has been done about the origins of Urdu and Hindi, no research study exists that compares and contrast Urdu and Hindi in a scholarly fashion (Russell, 1996). For quite some time, Urdu and Hindi were treated as the same language and, indeed, they are very similar. For geopolitical reasons, the languages can be classified into two languages. The geopolitical reasons are of no concern to this study, but they play an important role in why these two languages are currently diverging. Linguistic and pragmatic reasons are considered while studying the nature of Hindi and Urdu and their impact on computational processing. We will briefly broach the socio-linguistic aspects in terms of religion (i.e., Hindi for Hindus and Urdu for Muslims in India). The religious aspect

is relevant to understand their divergence from each other. This study intentionally doesn't dwell on this aspect too deeply but feel it is essential to bring it to the forefront.

Some experiments for computationally recognizing names show that Hindi and Urdu behave as two different languages. For example, lexical cues of recognition of locations are different. For example, *Dar-al-Khilafah* (Urdu) and *Rajdhani (Hindi)* are both used for the capitol of a city or a country. Therefore, more research is warranted to understand the relationship between these two languages to understand if the computational models based on one language can be used in some capacity for the other language. For this study, reasoning is based on the analysis of some of the revered scholars of the Urdu and Hindi languages, like Ralph Russell, David Matthews, Robert King, and Intizar Hussain. Then this reasoning is supported by empirical examples.

4.4.3 Hindustani is not the Predecessor of Hindi and Urdu

Some researchers claim that Urdu and Hindi emerged as the offspring of the earlier language known as Hindustani (King, 2001). Although there is some reference to a language called Hindustani in the later 1800s and early 20th century, a deeper analysis shows that no such language existed, and if it did, it was Urdu. Urdu is mentioned in South Asian literature as early as 1200 A.D and sometimes referred to as *Raikhta* (ریختا). Urdu was chosen as the official language of India by the British—that changed only after the Mutiny of 1857 (King, 2001) (Rahman, 2006). The use of the word "Hindustani" emerged from the leaders of the Congress Party, primarily Mahatma Gandhi, and then Jawaharlal Nehru, who wanted a common language for the United India. From the early 1900s, Gandhi relentlessly pursued the theory that the people of northern India spoke neither Persian-based Urdu nor Sanskrit-based Hindi; instead, they spoke Hindustani. The reason for this argument by the political party leaders was that there needed to be one language for the United India. Gandhi, and later Nehru, never talked about which script would be chosen because of the intense emotions attached to the issue (King, 2001). This political desire did not culminate as planned because most of the followers of the Congress Party wanted the Devanagari script to be the official script for Hindustani, an idea which was rejected

by the mostly Urdu-speaking populace in northern India[16] and Hyderabad Deccan. The use of the term *Hindustani* after the partition of India and Pakistan was used to describe the political tension around language choice during the British Raj.

4.4.4 Divergent Trend

Hindi and Urdu have two very clear differences: script and vocabulary. We will discuss the vocabulary differences in this section and script differences in section 4.6.5. Recent research in Urdu and Hindi studies is consistently arguing that Hindi and Urdu are two different languages and that they continue to diverge as time goes on. The most notable example of continual diverging is the ultra *sanskritizing*[17] of Hindi so much so that an Urdu speaker does not understand a Hindi news broadcast. This does not mean that Urdu and Hindi speakers don't understand each other; they do at an everyday level. Also, both Hindi and Urdu speakers who live together in Uttar Pradesh, Andhra Pradesh, and other large cities understand both languages effortlessly. But a person from an outside region who travels to these areas and knows only one language does not understand the other language in the streets, and broadcast media regardless of education level (Matthews, 2002), (King, 2001), (Russell, 1996). Matthews (2002) and Martynyuk (2003) make a similar comparison with Russian, Ukrainian, and Serbo-Croatian languages regarding their similarities in common vocabulary but still being different languages. King (2001) claims that the relationship between Urdu and Hindi is much more complex than Cyrillic orthography-based Serbian and Romanized Croatian collectively known as Serbo-Croatian. Given the discussion above, common vocabulary claim cannot be used to claim that two languages are same.

While analyzing the differences at a high level, they can be treated as the same language and play a pivotal role in establishing a link between South Asian communities around the world. A glowing example of this phenomenon is the Indian cinema where the line

[16] Delhi and Lucknow in northern India are considered birth place of Urdu

[17] Sanskritization is defined by anthropologist as spread of Vedantic and Brahmanical culture.

between Hindi and Urdu gets diluted. Although Hindi movies are popular in Urdu-speaking population and Urdu TV shows are popular in Hindi-speaking population, there is a steady and noticeable shift in Indian movies towards *sanskritizing* of Indian cinema. Remarkably, the Indian movies produced from the 1950s to the 1980s are undoubtedly Urdu (e.g., *Pakeezah* made in 1972 and Sanskritized *Swades* made in 2004). At a detailed level, Urdu and Hindi 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 (Matthews, 2002). Noticing the growing trend of the usage of Sanskrit words in Hindi, researchers of both Urdu and Hindi have started to describe them as separate languages. The commonality of the two languages is described by Matthews (2002) as an *unfortunate oversimplification* of two vibrant languages. Russell (1996) in his critique of Christopher King's book, "One Language Two Scripts," cites a examples where he shows that Hindi and Urdu are similar but different languages and sometimes the vocabulary, usage, and pronunciation can make a huge impact on the understanding of the language. Russell compared language teaching books of Hindi and noted that a number of the words could easily be treated as Urdu like *akela (alone)* and *akelapan (loneliness),* but soon the difference start to appear (e.g., *adhik (lots)* in contrast to *zyada (lots)* in Urdu*, akash (sky), asman(sky)* in Urdu).

A few examples like these should not be used to make a statement about two different languages. The translation of "How far is your house from here?" will be understood both by Hindi and Urdu speakers. But the divergence trend between these two languages continues with time (e.g., bicycle in Hindi referred to as *do chakr ghamia* which was noticed in Hindi news telecast whereas in Urdu it remains to be called *cycle*). The official usage is mostly present in the speech transcriptions of the Indian Parliament and the official government documents. The following example is borrowed from Russell to explain the growing divergence. 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".* Moreover, in

Hindi *"sadi"* could be replaced by *"satabdi"* and *"aur"* with *"tatha."* Russell points out that this example alone shows 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.

4.4.4.1 Highbrow, Middlebrow, and Lowbrow

Besides script, the most notable differences between Hindi and Urdu are found in the formalized vocabulary, grammar, and writing style. King (2001) quoting Ashok Kelkar, describes those differences in detail as an excellent example of a social linguistics situation. Hindi and Urdu have a full range of styles. He categorized those styles as stated below:

- *Formalized highbrow* is used in academia, religious sermons, official texts, and poetry. Most language engineering resources and enabling technologies for system development are based on this style. Highbrow Hindi draws its base from Sanskrit and highbrow Urdu, throughout time, has been based on Persian and Arabic words.
- *Formalized middlebrow* is used in songs, movies, pamphlets, popular printed literature, and mass propaganda.
- *Casual middlebrow* is most widely used for daily conversations among the educated upper and middle class who are regionally based like in northern India and Hyderabad. It is used for private communication like phone conversations and letter writing. It is used by newspapers so they can be read by a large audience, but the vocabulary in newspapers and media are diverging from each other, notwithstanding the script differences. This style is most receptive to borrowed words, most of them from English.
- *Casual lowbrow* is associated with what Kelkar calls the "lower class" and uneducated people. He calls it *bazaar Hindustani*. This is a substandard form of style. This style is found in the slums of urban centers of large Indian and Pakistani cities.

The polarization of Urdu and Hindi reaches its maximum at *formalized highbrow*. Hindi draws from Sanskrit for vocabulary and promotes Vedantic and Brahmanical culture. Urdu draws from Turkish and classical Persian literature and Islamic events as references.

King (2001) suggests standard Hindi (highbrow) and standard Urdu (highbrow) have diverged more since the partition of India and Pakistan in 1947. A careful analysis of King's theory shows that it is certainly true that standard Hindi is getting more and more sanskritized, but in general, new Urdu literature is leaning towards formalized middlebrow. Sanskritized Hindi is increasingly used by the elite and political figures in India. This movement of sanskritizing Hindi in India is illustrated by King (2001) while quoting Das Gupta and Gumperz. The illustration is done by analyzing the signboards; label *a* is the official text of the signboard, label *b* is the English translation, and label *c* is the causal middlebrow in Hindi. We have added label *d* as the highbrow in Urdu and label *e* is the middlebrow in Urdu.

- Signboard 1
  a. dhumrpan varjint hai
  b. smoking is prohibited
  c. cigarette pina mana hai
  d. tambakoo noshi mana hai
  e. cigarette pina mana hai

- Signboard 2
  a. Bina agya pravesh nishedh
  b. entrance prohibited without permission
  c. bina agya andar jana mana haid
  d. baghair poochey andar aana mana hai
  e. baghair poochey andar aana mana hai

Note that for signboard 2 middlebrow and highbrow is the same for Urdu as suggested earlier.

However, there is some evidence that the use of Arabic words is becoming prevalent in Urdu as Arab influence grows stronger in Pakistan. The following examples show the

trend as Urdu moves from Persian influence to Arabic. Label *a* is the original Urdu text, label *b* is the Arabic transformed Urdu text and label *c* is the English translation

- Example 1
  a. Kya tum ne *namaz* parhi?
  b. Kya tum ne *salat* parhi?
  c. Did you offer your prayers?

- Example 2
  a. Khuda Hafiz
  b. Allah Hafiz
  c. Goodbye – May God protect you

In example 1, *namaz* is universally understood in India and Pakistan as five daily prayers offered by Muslims. The introduction of the Arabic word *salat* is completely foreign to most of Hindi speakers. In example 2, the Persian word for God is substituted with the Arabic word for God. Although Hindi speaker will know what Allah means, they will be confused with the usage in context.

4.4.5 Cultural differences

Although languages don't belong to a religious group, it is an undeniable fact that Urdu is the first language of the large Muslim population in India and is known to most Muslim Pakistanis as their national language. The same is true for the Hindu majority in India where most people of Hindu faith in North India consider their first language to be Hindi. There are, of course, cross-pollination at the edges—Prem Chand and Gopi Chand Narang are notable examples.

The cultural preferences of speakers translate into respective languages. The date and year reference for Muslims for major events in South Asia and Muslim diaspora in the West is the Hijri calendar (a reference to Prophet Muhammad's migration from Makah to Medina). The year 2000 in C.E. is 1421 Hijri. This is evident in how different people reference the completion of the Taj Mahal for example. A Muslim cleric in Aligarh will

refer to its completion in 1076 Hijri, but a secular Hindu will say the date is 1666 A.D. The epitaph inside the Taj Mahal refers to the Hijri date, not the Gregorian calendar. Besides dates, the architectural terminology is quite different. A secular Hindu's description of the *Jama Masjid (*Central Mosque) in Delhi will be quite different than a local Muslim. For example, a secular Hindu might not be able to explain alignment of *mimbar* (a platform for the prayer leader) in the mosque or know that symbolizes direction to Makkah. Religious symbols can also be confusing. For example, a Muslim in Pakistan will not know the importance of the river *Ganga* for religious Hindus. *Mimbar* and *Ganga* are used considerably with their implied meanings in Urdu and Hindi literature.

Poetry and writings of scholars in the South Asia are highly biased towards Sanskritized culture in Hindi and references to Islamic events in Urdu. For example, epic elegiac poetry, the highest form of Urdu verse (Bailey 2008), about the tragedy of Karbala is called Marsiya. Marsiya is an essential part of Urdu literature where graduate degrees are awarded on their analysis where there are references to the Islamic events, and Quranic verses. Computationally analyzing these references for tasks like sentiment analysis in highbrow Urdu using Hindi bridge is not possible at this time because of the vocabulary divergence and cultural metaphors.

Urdu disseminated by networks of education and communication in colonial India after Persian was replaced by Urdu right after Mutiny of 1857. The Islamic seminaries used Urdu as a medium of instruction. Almost all Islamic theological seminaries and the Sufi poets wrote their literature in Urdu since 1857. Urdu was flourishing in the Mughal court before 1857. In fact, there was a considerable amount of Urdu poetry written by the last Mughal emperor Bahadur Shah Zafar. Urdu was one of the driving forces, next to Islam itself in the creation of Pakistan in 1947.

In India, Urdu symbolizes the Muslim identity of the Muslim minority against the domination of right-wing Hindu domination (Rahman 2006). This was most evident in the bloody riots in 1987 when Urdu was made the official language along with Hindi in Utter

Pradesh (a North Indian state) (King, 2001). In contrast, in Pakistan Urdu is supported by right-wing politics against English which is the hallmark of the elite in Pakistan.

4.4.6 Script Differences

In this section, I explain few of the script differences between Hindi and Urdu regarding phonemes, spoken units of a language, and graphemes, written units of a language. Hindi and Urdu have most of the phonological features of the languages of the subcontinent like retroflexion and voiceless and voiced, aspirated and unaspirated stops. The majority of the differences in Urdu and Hindi regarding the script are based on the vocabulary, where Urdu is prolific in borrowing sounds and words from other languages. Besides supporting the features of the languages of the sub-continent, the Urdu script supports the phonemes of Persian and Arabic. For example, in contrast to Hindi, Urdu has an unaspirated uvular stop /q/, labial fricative /f/, voiceless retroflex /ʂ/, velar fricative /x/, voiced dental fricative /z/, palato-alveolar voiced fricative /ʒ/, and voiced velar fricative /ɣ/. These sounds are supported by the nastaliq and naksh styles of Urdu script. Hindi has a system of making these sounds native by changing the articulation at different levels for each foreign sound. Urdu script has distinct graphemic features for retroflexion and aspiration. Urdu uses diacritic marks for retroflexion, and aspiration in Urdu is shown by *h* whereas the Devanagari script of Hindi does not treat retroflexion and aspiration as distinctive features. There are a number of other examples, but the few examples above justify the difficulty when using Hindi resources for Urdu computational processing. One of the easier tasks for language engineering is a transliteration from one language to another by using a map from one symbol to another. The work of Jawaid and Ahmed (2009) shows that there are many open issues when transliterating Hindi to Urdu or vice versa. The above differences show that Hindi stemmers cannot be used for Urdu stemmers (Riaz, 2007). The word *fiqh* (jurisprudence) in Urdu has labial fricative, uvular stop, and aspiration – all missing from Hindi script. A Hindi morphological analyzer or is most probably going to fail while parsing the token *فقھ (fiqh).* In another NLP task of transliteration *fiqha* (jurisprudence in Urdu middlebrow) will be transliterated to پھیکا

(bland) in Urdu and Hindi middlebrow. This shows that the differences in scripts have changed the meaning of the word during transliteration.

There are Urdu letters that have the same sound but are written differently. For example, ص، س، ث have the sound /s/ and ز، ذ، ض، ظ have the sound /z/. Hindi has these sounds but the mapping is one to one character to sound. Any machine, transliteration, machine translation, and NLP task will make errors while trying to resolve to the correct intent. For example, plenty of South Asian literature mentions offering goods to ward off the evil eye. Assuming the literature and tools are created for middlebrow Hindi and Urdu, consider the sentence:

Mother provided offering to ward off evil eye

ماں نے **نظر** اتارنے کے لئے **نذر** دی

In this example, **نذر***(nazr)* is *an offering* and نظر *(nazr)* means *evil eye.* Also, نظر *(nazr)* has an additional meaning of *sight.* Any NLP task using cross-language dictionary between Hindi and Urdu will have trouble getting the above sentence translated and transliterated correctly. Moreover, Urdu is bi-directional that is, both SVO and SOV, and therefore the above sentence can be written as:

ماں نے **نذر** دی **نظر** اتارنے کے لئے

The above discussion shows that scripts and phonetic sounds make bridging between two languages almost impossible with the current state of technology, and one language two script assumption does not hold

4.4.7 Quantitative Analysis of Hindi and Urdu

In this section, I show a few quantitative examples where Hindi resources cannot be used for Urdu Information Retrieval.

4.4.7.1 Named Entity Recognition

Named entity recognition (NER) is one of the important tasks in the field of data processing. It constitutes automatically recognizing proper nouns like people names, location names, and organization names in unstructured text. There has been significant

work done in English and European languages, but this task is not well-studied for the languages of South Asia.

By far the most comprehensive attempt made to study NER for South Asian and Southeast Asian languages was by the NER workshop at the International Joint Conference on 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. There are fifteen papers in the final proceedings of the NER workshop at IJCNLP 2008. A number of those papers tried to address all South Asian languages in general but resorted to Hindi where the most number of resources were available. There was not a single paper that focused on Urdu named entity recognition. In the papers that tried to address all languages, the computational model showed the lowest performance on Urdu. None of the researchers were able to use the online dictionaries and gazetteers that were available for Hindi for Urdu claiming the difference in vocabulary and the error rate in transcription due to phonetic and phonological differences.

4.4.7.2 Zipf's Law on Hindi and Urdu

One of the most interesting, and maybe the only, works available that compares Urdu and Hindi quantitatively is by Martynyuk (2003). In this study, Martynyuk establishes his argument based on the comparative analysis of the most frequent words used in various European languages. He then extends his work to compare Hindi and Urdu based on Zipf's Law. Zipf's law is one of the most fundamental laws used by researchers when analyzing text in an automated way is a foundation of Information Retrieval principles. On a high level, Zipf's law states that given a corpus of words, the frequency of the words is inversely proportional to the rank of the word. The hypothesis of the experiment is that if the most frequent words show the similar rank order in corpora of different languages like Hindi and Urdu, can be used as bridge language to solve many automated language processing tasks. Also, computational models derived for Hindi can apply to Urdu language processing. Hindi and Urdu news text tokens were categorized using manual

lemmatization. The experiment was conducted on approximately 440,000 Romanized words of Urdu and Hindi. After calculating the frequencies of the most frequent words, it was found that the top three words between Urdu and Hindi were the same words (stop words). The rank order between the words starts to divert after rank three. The top 24 ranked words are the same between the two languages but have a different rank order than each other. After rank 24, the words start to differ in rank order and their alignment with the corresponding word from the other language. For example, the Hindi word for election, *chunaav,* is at rank 25; the Urdu equivalent for election, *intikhabaat,* is found at rank 45. It is important to note that the Hindi corpus did not contain the English word *election,* but in the Urdu corpus it was used almost as many times as *intikhabaat* (*intikhabaat* is used 672 times, and *election* is used 642 times.) This next example may drive home the point that the rank order of the same meaning words keeps on drifting apart. The word for terrorist in Hindi, *atankvadee,* is at rank 42 in contrast to the equivalent word in Urdu, *dehshatgard,* which is at rank 182. This experiment clearly shows the given parallel corpus, Information Retrieval features like term frequency, inverse document frequency, and stop word analysis for Hindi and Urdu will show different results because of the difference in top-ranked words (Riaz, 2007).

4.4.7.3 Hindi Word Net

WordNet is an important enabling technology for conceptual understanding and word sense disambiguation tasks. It is a critical tool for low resource languages for term expansion in Information Retrieval. Hindi Wordnet (Jha et al., 2001) is an excellent source for Hindi language processing but cannot be used for Urdu. Most of the analysis of the words and the categorization of words in Hindi WordNet is done by using highbrow Hindi. For example, the terminology used to describe parts of speech (POS) in Hindi Wordnet is completely foreign to Urdu speakers. The POS names are Sanskrit-based whereas Urdu POS words are Persian-based and Arabic-based. In Hindi the word for the noun is *sangya* and in Urdu it is *ism.* The proper noun in Hindi is called *Vyakti vachak sang.* No Urdu speaker would know this unless they have studied Hindi grammar. To work through these differences, one has to be familiar with both languages at almost expert levels.

Ahmed et al. (2010) tried to create Urdu Wordnet using state of the art transliteration between Hindi and Urdu. Here the authors claim that the two languages have the same vocabulary and therefore the Urdu words will be matched in Hindi Wordnet. First, they used the lexical structures (c-structure) and found it to be too error-prone, and then used (f-structure) to assign grammatical relations. This approach works well for basic simple sentences like *he ate an apple*, while assumeing that Urdu words will be present in Hindi WordNet. We assert that this is not a valid assumption when working with news or literary material because of diverging vocabularies as reasons described in earlier sections. Consider the Urdu word for *dawn* is سحر pronounced as *Sahr*. All the phones of this word are present in Hindi, and we assume that the transliteration will be correct in Hindi which should be सहर. When this term was searched in the Hindi WordNet, there were no matches.

Consider a situation where there are multiple words in Urdu that sound the same but have different meaning, and their initial letters are different in Urdu. An example is صراب (syrup) with the initial letter of ص, and سراب (mirage) with the initial letter of س, both have the sound of *Saraab* so they are considered homophones of each other. Hindi WordNet does not have equivalent words and if the grammatical relationship like Lexical Functional Grammar (LFG) is used as suggested by Ahmed (2010) it will not be able to disambiguate because of the roman transliteration errors because Hindi does have a letter that distinguishes between س /s/ and ص /s/. Let us assume if there is some inference matching engine like soundex to see the closest match in Hindi WordNet, the word could become शराब *Sharaab* which means alcohol in both Urdu and Hindi with incorrect semantic interpretation in case of صراب (*syrup)*.

Consider the table below where some arbitrary words are chosen mostly from middlebrow diction using newspapers. It is evident that vocabulary mismatch, along with the phonetic differences, majority of the hits on Hindi Wordnet will not return any results.

Urdu Word	Roman	Hindi	Roman Hindi	In Hindi WordNet
صراب (Syrup)	Saraab	सराब	Saraab	No
سراب (mirage)	Saraab	सराब	Saraab	No
مینہ (alcohol)	Mein	मेंह (Rain)	Mehn	No
مینہ کدہ (bar)	Mein Kadah	मेंह कदह	Menh Kadah	No
ذھر (poison)	Zahr	ज़खर	Zakhar (incorrect transliteration)	No
زہر (poison)	Zahr	जहर	Jahar	Yes

Table 11

It is not implied that Ahmed (2010) study is of no value. It is a step in a right direction and will work with simple sentences. Their assumption of one language and two scripts will hold to create a knowledge resource for Urdu because of error rate and vocabulary mismatch. Therefore, Urdu computational tasks like Information Retrieval, NLP, and NER will be suboptimal for practical use.

4.4.7.4 Machine Translation

Recently, there have been attempts to translate English to Urdu through Hindi since Urdu does not have extensive resources like online dictionaries. Mahesh (2009) used handcrafted mapping tables from Hindi to Urdu and used existing English to Hindi translation engine to come up with a translation. The translation engine's results when reviewed by a native Urdu speaker had a problem in every sentence. Problems were in

case marking, plurals, and incorrect grammatical construction of the constituent. Each Urdu example had incorrect Urdu orthography. Although the study shows some promising results, the results show that the difference between the two languages is more than only the script. The results are far from acceptable because Hindi resources do not reflect the vocabulary usage in Urdu sentences the accurately.

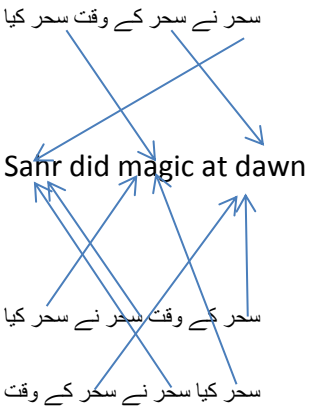
Visweswariah et al. (2010) also assuming one language two scripts principle addressed statistical Machine Translation and other NLP tasks for Urdu and Hindi in the absence of Urdu-Hindi Parallel corpus and Hindi-English parallel corpus. First, they acknowledge that machine transliteration between the two languages is riddled with problems. Also, they estimate 28% of Urdu tokens will not be natural in Hindi assuming perfect transliteration. I suspect this number to be quite high because of high error rate in transliteration. Visweswariah et al., also assume that in Urdu-Hindi translation word order does not change except in a few situations—the example below shows otherwise. They use English as a pivot language to generate conditional probabilities of Urdu words generated by a Hindi word supported by the Hindi-Urdu-English dictionary. Statistical alignment approach suffered from errors in word alignment, tokenization errors, and handling of rare words which were accommodated by hand created lookup tables. The authors assume that named entities which are rare will align very well so a transliteration approach will work well. This is not a correct assumption as described in section 4.4.5. and shown in section 4.4.6 where an Urdu script named entity has with unsupported sounds or letters in Hindi. Consider following sentences in Urdu. Sentence (a) is written as SOV, sentence (b) is written as OSV, and sentence (c) is written as VSO The first two are colloquial and will be found in newspaper equally likely. The (c) construction is unlikely in a narration but very likely in a poetic literature of Urdu where a writer wants to stick with a rhyme scheme.

a) سحر نے سحر کے وقت سحر کیا

Sahr did magic at dawn

b) سحر کے وقت سحر نے سحر کیا

c) سحر کیا سحر نے سحر کے وقت



The word Sahr has three meanings, *PERSON*, *magic*, and *TIME* in NER tag scheme. We use arrows to show the polysemy of the word Sahr as a person, time, and magic. The NLP components used in this study like MaxEnt word aligner and POS tagger will run into resolution errors because it will not know by looking at the static list if Sahr should be resolved to time, magic or a person. Also, none of these words appear in a Hindi WordNet to resolve word sense.

4.4.8 Summary

In conclusion, assert that Hindi and Urdu are two different languages, at least for computational processes. There are significant differences in the vocabulary, scripts, phonetics, and phonology that they cannot be used interchangeably in order to do NLP tasks. We assert that the current tools available for the Hindi language cannot be used for Urdu language processing at a large scale for deeper semantic analysis. In other words, to use Hindi resources to do Urdu computational processing, one has to know Hindi at a detailed linguistic and cultural level. The examples of the quantitative analysis confirm the emerging research from the Urdu and Hindi language researchers that there is a trend of divergence between the two languages because of the vocabulary usage. There is a great opportunity to use casual middlebrow where they can mutually intelligible to bridge these languages and develop tools for multilingual information processing. However, linguistic resources of Hindi are created for highbrow usage. A few examples of NER, transliteration, and machine translation show that there is increasing gap between these two vibrant languages. The diverging trends of these languages and the polysemy found

in Urdu direct us to create new computational resources in Urdu computational processing

4.5 Urdu Named Entity Recognition

The goal of NER task is to automatically identify the boundaries of a variety of phrases in raw text and then to categorize the texts identified. There are three categories, TIMEX to recognize time, NUMEX to recognize numbers, and ENAMEX to recognize proper nouns. ENAMEX can be further categorized based on a domain. The NER task tries to categorize ENAMEX entities into geographic location, company names, and people names to name a few. This research in Urdu NER focuses on proper nouns, geographic locations, organizations, date, and numbers in Urdu text. 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.

This section describes the one of the main contribution of this research – Urdu NER system and the Urdu NER algorithm, and challenges in creating it. Urdu is vastly spoken in South Asia so it can be considered as one of South Asian languages. Urdu is considered on of the lowest resource languages for NER because it does not have a number of linguistic resources that are used to boost the performance like gazetteers, look up dictionaries etc.

4.5.1 NER for South Asian languages and Related Work

One of the major reasons for the lack of research is the lack of enabling technologies such as 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 Riaz et al. (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 (Becker and Riaz 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. A number of approaches for name recognition were tried with variant results. 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 of the pile with the least amount of training 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 IJNLP 2008, a number of those papers tried to address South Asian or Indian languages in general, but resorted to Hindi, where the most number of resources were available. A number of papers only addressed specific languages like Hindi, Bengali, Telugu and one paper addressed Tamil. There was not a single paper that focused on Urdu named entity recognition. The papers that tried to address all languages, the computational model showed the lowest performance on Urdu, the references of these papers are available in the bibliography. The following section describes some of the research challenges when trying to do NER task in Urdu. The challenges provided are specific to NER.

4.5.2 NER challenges for Urdu

In general NER is a difficult task and a number of challenges need to be addressed in all languages. The causes of these challenges are due to the unique nature of Urdu, i.e. it is an MRL, written in Arabic Script, does not have a word order constrant, an Indo Aryan language, that borrows heavily from other languages resulting in polysemy to name a few. Although some issues are common to other Indo Aryan languages, the focus is on language characteristics and some practical problems of Urdu NER. These issues don't occur orthogonally that is an Urdu word could exhibit a number of these challenges.

4.5.2.1 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. Urdu

orthography does not support capitalization and therefore, this important feature cannot be used for Urdu and sometimes degrades the performance of the system. English systems easily recognize acronyms by using capitalization, but in Urdu they are quite difficult to recognize. For example B.B.C in English vs. بی بی سی in Urdu.

4.5.2.2 Agglutinative nature

Agglutinative nature means that some additional features can be added to the word to add more complex meaning. This phenomenon is mostly observed in Telugu and Urdu. 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 and Turkish. In Urdu the *Hyderabad* + *i, = Hyderabadi* حیدرآبادی. Here *Hyderabadi* should not be recognized as a proper noun whereas *Hyderabad* should be recognized as a location named entity. This particular issue can cause degradation in Search and NER performance. During the search evaluation Urdu and analysis of NER systems in Persian if اسرایلی /Israeli/ is stemmed to اسرایل/*Israel*/ Search and NER performance suffered.

4.5.2.3 Ambiguity

Ambiguity in proper name names is also present in Urdu as in English. The names like Brown are ambiguous in English – name or color. Similarly, Sahar is ambiguous in Urdu – name or morning dawn. In Urdu this gets more complicated because the Sahr also means casting a spell. An interesting sentence in Urdu can be *Sahr ne Sahr ke waqt Sahr kia* which says *Sahr casted a spell in the morning.*

4.5.2.4 Proper and Common Noun Ambiguity

Sometime the common nouns can be used as proper names in Urdu. An example from Hindi and Urdu vocabulary is *rakhee (a cloth hand bracelet),* and name *Rakhi.* The previous section shows an example in Urdu where *Sahr* is a name, spell, and dawn.

4.5.2.5 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. For example, in news text the word order is consistent in the body of the news but the title of the news story does not respect word order. The titles of the news text often are scribed in the word order that is "improper" for Urdu because of emphasis. Urdu is considered free word order language 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.*

4.5.2.6 Spellings

A number of situations occur in news articles where different authors or news reporters scribe the name in different spellings even for native Urdu names. In English, this is caught by capitalization, Soundex like algorithms, and co-reference resolution but in Urdu in the absence of capitalization this becomes a problem. Soundex other state of art fuzzy matching don't work for Arabic Script languages. An example is مسعود and مسود represents the name Masood where co-reference resolution is one way to solve it.

4.5.2.7 Ambiguity in Suffixes

A very common phenomenon in the proper names and common names in the Urdu is the use of a location suffix to in a person name. Sometime the suffix is attached to the location name like a building or road. A common practice in the Punjab region of the Indian Subcontinent is to append the location of person's origin in their name with a suffix *i* or *vi.* For example, if the person or their ancestors were from Balta (a city of Punjab), *vi* is added to the name to form *Batalavi.* If the person is from Jalandhar, the suffix *i* is added to form Jalandhari. Sometimes the name of the location ends with phoneme *i,* in this case a word like *wallah e.g. Mianwali wallah.* Besides suffixes, Urdu poets use a pseudonym of their choosing at the end their name called *Takhalus. Gahlib* is the takhalus of the famous Urdu and Persian poet Asad Ullah Khan *Ghalib*.

4.5.2.8 Loan words in Urdu

Urdu has the exception of all other South Asian languages to contain 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, *bin* an Arabic cue need to be used in the middle of the name for the person name.

4.5.2.9 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 English text example will be *John F. Kennedy Airport*. An example in Urdu news text is *Punjab University* where *Punjab* is the location name and *University* marks the whole entity as an organization.

4.5.2.10 Bidirectional nature

In contrast to all other South Asian languages Urdu is written in the Arabic script and it is written right to left while the numbers go from left to right. Since Urdu vocabulary consists of a myriad of languages including English, it is not uncommon to see English words written in Latin script from left to right. For example, وہ میرا laptop ہے is Urdu equivalent of *That is my Laptop.* After carefully analyzing Becker-Riaz corpus's top 2000 documents the bidirectional was present mostly when scribing numbers, and dates.

4.5.2.11 Conjunction Ambiguity

Urdu text shows quite a few examples of conjunction ambiguities among proper nouns. English equivalent is Toyota and Honda Motor Company. Although this phenomena is present is most languages, none of the papers in IJCNLP NER workshop mentioned them as a problem. In English, capitalization can address this issue along with world knowledge. An example of conjunction ambiguity in Urdu for Becker-Riaz corpus is مشرف اور شجاعت نے کھانا کھایا *(Musharraf and Shujaat ate dinner)* where Musharraf and Shujaat are political figures.

4.5.2.12 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 and Becker-Riaz (2002) corpus. The EMILLE corpus contains long running articles that don't 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 IJNLP 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 almost all of all experiments used statistical machine learning for name recognition and conditional random fields 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 almost all researchers claimed the lack of the other resources to boost the 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.

Among the experiments performed at Named Entity Workshop on various Indic languages and Urdu, almost all experiments used CRF with limited success. The workshop provided training data for all the languages but all better performing systems used their own tagged resources from various Indian institutes. In addition, all systems tried to use gazetteers to boost the performance. In short, none of the systems that used state of the art Conditional Random Fields had good results when tried on South Asian languages in contrast to English. Moreover, analysis of other morphological resource languages showed that all of them used gazetteers. The statistical approaches required high quality

annotated training data, in the absence of it the system performed poorly. Shalaan(2014) suggests after studying Arabic NER algorithms that rule based approaches are most suited for MRL languages because in order to represent most features of an MRL a large annotated corpus needs to be created with features annotated correctly.

4.5.3 Rule-based Urdu NER

We used a handcrafted 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 the art for named entity extraction, in the absence of boosting methods like gazetteers, CRF performed poorly with only annotated text.
- Deep learning methodologies, although popular, require massive amount of annotated corpus and word embeddings. There are no large annotated corpora for Urdu to learn features automatically. Furthermore, LSA word embeddings have are suboptimal as shown in chapter 3.
- 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).
- To create a new set of tagged data set for modeling CRF or other new statistical algorithm on Urdu data is cost prohibitive at this time.
- Morphologically Rich languages results of Farsi, Arabic, and Russian showed that rule-based methods are much more robust, the features required for statistical machine language learning like CRF are too complex to create. For example, using features through clustering will not help because of polysemy and homonymy (Shalaan 2014).

4.5.3.1 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>نہیں 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

کرنے کی راہ ہموار ہو جائے گی۔ تاہم امریکی سینٹ اب اس استعمال کرنے کی راہ ہموار ہو جائے گی۔ تاہم امریکی سینٹ اب اس معاملے پر غور 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. Some interesting ones below:

- Punctuation marks like ":" are useful, but the position of their occurrence in the text is important.
- Beginning of the sentence in the title of news text has a different rule than the 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 the sentence. Moreover, POS tagger changed the order of words. This further complicated writing matching rules. Hence available POS taggers are not helpful and harm Urdu processing.
- Stemming reduces the precision of the system. It will conflate terms like *Pakistani* to *Pakistan.* Hence, marking *Pakistan* as a 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.
- The 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. Identifying such entities may require building a Knowledge Graph
- Co-reference resolution for names will be non-trivial since they have multiple spellings, only the context can be used to resolve them. For example, *Milosevic* is spelled at least with three different spelling in Urdu sample set.

- Honorific titles are very important, but a title like *Sadr (President)* can occasionally lead to incorrect recognition because *Sadr* is the location of a neighborhood of *Karachi (largest city in Pakistan).* Moreover, *Sadr* is the last name of a revered Iraqi scholar active in Middle East politics. The Urdu NER algorithm adjusts to such situation.
- 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 and is empirically chosen which gave the best performance.

4.5.3.2 Algorithm for Urdu NER

In the Urdu NER rule-based system for this research, the rules form a finite state automaton (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, linguistics driven heuristic-based, and Urdu 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 a 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 a regular expression 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 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 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* frequently occur 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 the 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 named 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. 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 the subsequent 6-grams.

The rules at the top of the list could tie for importance, e.g., The rules for جنرل فیصل *(General Faisal)* and فیصل شاہراہ *(Shahrah-e-Faisal or Faisal Boulevard)* have very consistent previous token cues. The 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 فیصل شاہراہ as the location if the *"general"* rule was applied first. This has the side-effect of low recall for nested-entities.

4.5.3.3 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 the 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 of names 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 newsworthy 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%. Suffixes cues and anchor text features were very useful feature but at the same time anchor text feature was the cause of 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.

4.5.4 Discussion

Although 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.
- Case markers or Urdu postpositions suffix are used as 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.
- Some rules were removed where an 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 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.

4.5.5 Summary

Urdu NER is a significant intellectual contribution in this research. This research shows that rule-based approach for a resource lacking language is quite effective on an MRL with low resources. Urdu NER does use ad-hoc heuristics but base rules using the Urdu

grammar's features of lexically independent case markers, postpositions, Urdu morphology, and selected stop words based on the algorithm presented in the stop word section (Riaz 2007). This strategy is much more effective than a statistical CRF based model without gazetteers and shows significant improvement over statistically based baselines.

4.6 Chapter Summary

This chapter showed that research in low-resource MRL is quite challenging as resources among MRL are non-sharable. Special care needs to be taken to create tools and algorithms for computational tasks like a corpus, stop words generation, stemming, base-line creation and named entity recognizers.

The next chapter examines the NER approaches for other MRL in this dissertation, and provides analysis in comparison with the Urdu NER algorithm.

5 NER Analysis for Other Morphological Rich Languages

There has been considerable research done on named entity resolution (NER) for languages that are resource-rich and non-MRL. In this chapter, NER approaches for resource-scarce MRL are evaluated. The example languages are Farsi and Russian, and Arabic. Among these MRL, Arabic is the most studied with some linguistic resources available, but these resources lack quality (Shalaan 2014). State of art NER algorithms are examined for each language in detail, and an attempt is made to replicate those approaches with Urdu if such resources can be publically available. In the analysis section of each language, the effect of each approach is explored for Urdu with examples to show the similarity and differences of MRL.

5.1 Farsi / Persian

Persian belongs to the Indo-Iranian branch of the Indo-European family. There are three variants of the language: Western Persian, referred to as Parsi or Farsi and spoken in Iran; Eastern Persian, referred to as Dari and spoken in Afghanistan; and Tajiki, spoken in Tajikistan and Uzbekistan. Persian has also had a strong influence on neighboring languages such as Turkish, Armenian, Azerbaijani, Urdu, Pashto, and Punjabi. Persian is a more expressive language and more difficult than English for computational processing because of complex morphology (Khormuji 2014).

5.1.1 Introduction

Persian is a resource-scarce language; there are very few public resources or tools available for language processing task. The first linguistically annotated Persian corpus was the Bijankhan Corpus (Bijankhan, 2004) released in 2004. There are no publically available training data or software components for Persian NER. Literature review shows two Persian NER systems; one is by Khormuji (2014) based on dictionary lookups and the other by Farhadi (2014) based on web-based language translation, and using Web-based English language NER systems to create an ontology.

Since the last century, Persian names are composed of a first name and a last name. There are no middle names. The first name and the last name could be formed of other subparts.

The first names are mostly of Arabic or Islamic origin, and the last names usually have a common prefix of -i- or ی For example, a current political figure Ali-Akbar Vilayati علی‌اکبر ولایتی (Megerdoomian, 2008).

Like Urdu, there is lack of capitalization, the absence of short vowels, and there is optionality in spacing. Unlike Urdu, Farsi has determiners that can be used to distinguish proper nouns and common nouns.

5.1.2 Approaches

Khormuji (2014) refer to their NER system as DB-NER because it is based on database lookups instead of annotated training data. DB-NER is similar to English name screening commercial systems used by government agencies and global banks to identify suspect entities. Their dictionary based NER framework is based on the first creation of dictionaries and then a lookup mechanism for input text to detect candidate named entities. The candidates are filtered to remove false positives. Khormuji et al.'s approach is described below.

- **Dictionary creation:** The most fundamental component of DB-NER is the dictionaries. There are multiple dictionaries prepared for DB-NER. The source data for dictionaries to populate is stored at National Library and Archives Organization of Iran (NLAI). Since the data stored at NLAI is stored in RDF triples and identifies classes like a *person, location, organization, and article title,* some other derived dictionaries were also created. The DB-NER have the following dictionaries. *First names, Last names, Full names, location, organizations, miscellaneous,* and *products.*
- **Preprocessing:** Both *input text* and *dictionaries* go through preprocessing. The input text preprocessing is required because most dictionary entries don't match the input text. Preprocessing of input text consists of language dependent tokenization of words, punctuations. The date formats are preprocessed to differentiate between Hijri date format and Gregorian date format. Dictionary preprocessing is required because the entries in the dictionary are noisy. Preprocessing entails removing

phrases with stop words and efficient storage mechanism to aide is matching text and dictionary entries.

- **Candidate detection:** Candidate detection is a matching process where the input text is matched with the dictionary entries. Three matching strategies are used for experimentation: a) longest substring match, b) exact match, and c) fuzzy match. For each of the matching strategies, the input string is matched with and without stemming.
- **Candidate filtering:** Despite removing noisy phrases, the noisy data remained and produced a large number of incorrectly named entities. Parts of speech (POS) taggers are used to identify false positives.
- **Post processing:** POS taggers make a number of mistakes and can't recognize the difference between proper noun and noun. Another filter is introduced that explores the dictionary to determine if nouns are in the dictionaries.

Farhadi's (2014) primary goal is not to recognize Persian named entities in native Arabic script but to create an ontology of Persian named entities that is enriched by semantic features. These features can be integrated into the current Web-based ontologies. The system is referred as **NER-FL** by the author. The input to the system is native Persian Arabic script text that goes through stop word removal process and fed into Google Translate system. The English translation is fed into Web-based commercial text processing system to identify named entities automatically. An NER-FL ontology schema is created to merge the ontology results for various NLP systems, and output of these NLP systems are mapped to create an NER-FL ontology. The contribution of this work is to enrich NER-FL ontology in a news domain represented in English. This approach does not process Arabic script Persian language for named entities.

5.1.3 Corpus

Khormuji et al. used Bijankhan Corpus as an input text to match with the dictionary entries. The corpus was created from online material consisting of texts with different genres and topics such as newspaper articles, fiction, technical descriptions, and writings

about culture and art. The corpus contains 2,597,939 tokens and is annotated with morphosyntactic and partly semantic features. The corpus contains both English and Persian content. Some elements of the corpus ecosystem are available to download. Although the authors reported that the corpus and related resources have annotated information about named entities, there is no such resource available publicly or commercially per communication from the stewards of the data. The only named entity related resource available is a file which indicates that it has named entities and their frequencies. This file contains English and Persian entries. Analysis of this file shows that it contains frequencies of nouns and names and in addition to other tokens. An excerpt of the file *BijanKhan_Word_Sorted_Name.txt* is provided below in two tables – English and Persian shows there are not named entities.

English	**Word Count**
frmDataType	1
frmImpTbls	1
from	1
fromdb	2
fromtable	2
ft	4
function	1
g	1

Table 12

Excerpt showing Persian entries

Persian	Word Count	Translation
آب	1455	water
آب آورده	1	brought water
آب اسك	1	water ski
آب اكسيژنه	1	Hydrogen Peroxide
آب انبار	33	Cistern
آب انبارها	1	Water cisterns
آب انبارهاي	4	Water storage
آب بازي	1	Water play

Table 13

Farhadi (2014) used 1200 news articles from the Persian language version of the Fars News Agency ranging a month in 2014

5.1.4 Evaluation

Khormuji et al. tried a number of configurations for matching and found that DB-NER achieved best F1-measure with Proper Noun filtering when both dictionary entries and text input were stemmed. Exact phrase search and fuzzy match search reported F1-measure of about 81%. The exact matching without stemming regardless of filters reported F1-measure of 42%.

5.1.5 Analysis

DB-NER is not a machine learned system for NER instead it is a string match or a “search” based solution. The results clearly show that language based processing like stop word removal, tokenization, and stemming are necessary to improve the accuracy of an NER using Search. This is not a surprise since DB-NER is more like a string matching solution

and stemming and stop words removal should improve recall. The rule-based system of filters helps to improve precision. The Urdu NER system developed for this dissertation does not use dictionaries because the considerable manual effort it will require to annotate the content, and most importantly there are no sources of Urdu repository like National Library and Archives Organization of Iran (NLAI) for Farsi.

NER-FL processes the Arabic Script Persian language for UI purposes, and to generates stop words for its trimming module. The stop words generation process is similar to Riaz (2007) and detailed in section 4.2 where the most frequently used words in the corpus are considered stop words. The mining systems used by NER-FL Text Razor, Alchemy, and DBpedia spotlight. These systems work well on English language data but do not support Urdu data per mine evaluation. Experiments using Urdu text on these systems resulted in error messages that Urdu is not supported. Urdu text when processed through Google Translate to simulate this experiment produced approximate but not correct translations. For example:

Urdu: پاسدرانِ انقلاب پر پابندی کا منہ توڑ جواب دیں گے: ایران

English1: Iran will be able to break the ban on the neighboring revolution

English 2: Pasdran revolution will give a befitting reply to ban Iran

The Urdu sentence above is taken from BBC Urdu, and it translates as:

Iran: There will be serious retaliation for a ban on the Revolutionary Guards.

It is evident that the translation is somewhat incorrect for the first English translation because it failed to recognize the object—a named entity. The second translation is completely wrong by Google Translate and provides the exact opposite meaning of the sentence. In the Urdu sentence, the phrase پاسدرانِ انقلاب *(the Guardians of Revolution)* or the *Revolutionary Guards* is a Persian phrase construction and a proper noun referring to Iran's elite force *The Revolutionary Guards*. This Persian phrase construction used natively in Urdu and is called Izafat as described in section 4.2.3.1 as a challenging linguistic feature to process for MRL languages like Urdu and Farsi. The above discussion shows that given

the available translation technologies, natively processing content in its original script is less error-prone than using a translation based mechanism. It also shows that the challenges that are associated with robust and error-free linguistic resources for NER.

5.2 Russian

Russian is a Slavic language of the Indo-European family. It is an official language of Russia, Belarus, Kazakhstan, and Kyrgyzstan. It is widely spoken in Ukraine and Latvia. Some former Soviet states have given it a status of a minority language. It is spoken by a large number of world population and by some estimates ranked 8th by the total number of native speakers. It is written in Cyrillic script and is considered highly inflectional. The Russian language has capitalization to indicate proper nouns. The spoken Russian language, as Farsi and Urdu, is influenced by the literary language. And like most Slavic languages, Russian uses prefixes and suffixes to build vocabulary.

5.2.1 Introduction

Despite a large number of Russian speakers, there are hardly any resources available for Russian NER Gareev(2013). Most of the research and computational resources are available for machine translation due to the cold war. Although name identification is considered an important pre-step for Machine Translation, there is no significant work to identify Russian names. Literature review shows Gareev et al. (2013) to be the first ones to introduce a baseline for NER approaches and a corpus for NER.

The Russian naming system is significant not only because of its use throughout the Russian Federation but also because of the influence it has had on naming conventions in other regions of the world. As a result of the vast expansion of the Russian empire and later the Soviet Union, the Russian naming system was introduced to many other cultural areas, including some outside of the Slavic region. Since the dissolution of the Soviet Union, many former Soviet republics have taken steps to return to their traditional naming systems. Because of this process, notably, in Central Asia, it is not uncommon to have two variants of established names. (Lisbach, 2013)

Traditional Russian person names are made up of three parts: a) the given name b) patronymic name, and c) family name. For instance, the full name of the great Russian novelist Tolstoy is Lev (given name) Nikolayevich (patronymic) Tolstoy (family name). The patronymic is based on the first name of a person's father, adding either a masculine (-vich or -ich) or feminine (Ruff, 2015). The given names often have diminutive forms which are frequently used for good acquaintances. These forms usually have no obvious connection to their related full name. For example, a diminutive form of Vladimir can be Vova, Volodya, or Vovochka and the diminutive forms of Aleksandar can be Sasha, Sanya, Shurik, Alik (Libach, 2013).

5.2.2 Approach

Gareev et al. (2013) introduced baseline for Russian NER that comprised of:

- a dictionary lookup knowledge-based approach for Russian NER
- a statistical based NER approach for Russian NER
- an evaluation and test corpus for Russian NER

The knowledge-based approach is based on dictionary lookups, pattern-based and looks for named entity variations in a single document.

- **Knowledge-based approach**

For a dictionary-based approach, several named dictionaries were prepared.

- Five organization dictionaries were assembled, four Russian dictionaries and one English. The first organization dictionary (D1) was sourced from the Russian government registry. These entries were post-processed to remove single entry ambiguous and adding more information to legal entities, so they are not ambiguous like adding “Corp.” to a corporation. D1 has 2.3 million entries. The second list of organization (D2) is sourced from Russian DBPedia. D2 has about 40,000 entries. The third list of organization (D3) is the augmented D2 with the anchor texts from DBpedia Spotlight anchor texts. D3 has about 87,000 entries. The fourth organization dictionary (D4) was created from the newswire’s company catalogs like

https://expert.ru/dossier/companies/. D4 contains about 280,000 entries. The English organization dictionary (ED) was created from English DBPedia. D5 has about 280,000 entries.

- The authors did not provide the details of the source or the methodology of creating Person names dictionary. It contains 10,000 male and 10,000 female names.

The dictionary lookup for entities can be quite noisy because the input text can be different from dictionary text regarding inflection or in case sensitivity. Stemming and case sensitivity can be turned off while matching input to the dictionary entry. The dictionary lookup only matches organization and first names.

The pattern-based approach uses the context cues to recognize the noun phrases that denote the entity type. There are two types of patterns, first-class patterns that indicated a proper name with confidence and some that are possible candidates. The possible candidates are resolved at the document based entity resolution step. These pattern indicators were created manually during the D1 dictionary creation process. For example, *ООО/Ltd., фонд/foundation,* etc. There are 2,700 organization indicator patterns and 50-person name indicator patterns. The person name match is a simple look up of text with a capitalized version word in the Person dictionary.

The Document based entity resolution is used when the first mention is full entity name, and subsequent mentions are short forms. A list of candidates that have short forms are recognized at the pattern matching stage, and their resolution is done within the document scope. All the possible short forms are evaluated by a voting mechanism to see which pattern is fired. The short form entity that gets the most votes is assigned that entity type.

- **Statistical Approach**

Like most NER systems based on statistical approach, Gareev et al. used linear chain Condition Random Fields (CRF). The authors used CRF based on publically available Mallet system with 5-fold cross-validation. The authors realize that the corpus is not big enough to be trained reliably. The features used of the CFR based system are quite elaborate that

include the features from other statistical and cluster-based features. Some of the notable features are:

- Current token
- 5 token window centered on the current token
- The shape of the letter around the current token
- The prefixes and suffixes of the current token
- LDA cluster labels
- Numerous cluster generated tokens

5.2.3 Corpus

This first Russian NER corpus was introduced by Gareev (2013) along with the baseline that includes names and organization mentions. The source documents are ten newswires business feeds from Yandex. The corpus contains ten news articles from each of the feeds that were collected on a single date. The post-processing of corpus de duplicates news articles and removes title -- the corpus contains only the body of the news article. There are a total of 97 documents available in the corpus that are manually annotated. The authors don't mention how many or if any cross agreement were measured. The corpus contains annotations of person names, and organizations only. Only the direct mentions are annotated that is, any anaphora references are not annotated. Also, nested entities are not annotated, for example, either Microsoft is annotated or Microsoft Office.

There are a number of articles in the corpus that don't have Russian named entities and therefore, only the English named entities are tagged. Each token is represented on a separate line. For example, National Hurricane Center is represented as:

Национальный	B-ORG	National
центр	I-ORG	Hurricane
ураганов	I-ORG	Center

Table 14

The English entities are tagged as:

S B-ORG

& I-ORG

P I-ORG

Table 15

Although not explained in Gareev (2013), the analysis of the corpus and tagging instances show that the annotators mark the beginning of the organization entity (B-ORG) and the all the internal tokens it comprises of is marked with I-ORG as an internal entity. The absence of the entity tag in the next token indicates the end of an entity. Annotator used the same scheme for person names also. Any Latin characters that represent a name but are not capitalized are not tagged as entities, e.g., Cameron. The corpus statistics are shown below:

Tokens	44326
Organizations	1317
Person	486

Table 16

5.2.4 Evaluation

5.2.4.1 Knowledge-based approach

The evaluation of the knowledge-based approach showed a number of dictionary coverage and quality issues. The person name recognition constituted a simple lookup to the Person dictionary reported F_1-measure of 68.64%. The person false negatives are caused by incomplete dictionary of first names. Incorrect stemming for first and second names and the weak coverage of person indicators in pattern matching also cause person false negatives. Futhermore, false positive in person names is due to the ambiguous

patterns. The English dictionary (ED) lookup without stemming reported F_1-measure of 18.9% The performance of Organizations is provided below with each configuration.

Configuration	**F-measure**
English (ED) only without stemming	18.6%
English (ED), and D1, stemming	9%
English (ED), and D2, stemming	44%
English (ED), and D3, stemming	35%
English (ED), D4, stemming	38.5
Pattern Matching	29.50%
Pattern Matching and Doc Analysis	54%
English, Doc Analysis, Pattern Matching, newswire catalog lookup	55.48%

Table 17

The error analysis shows that D1, D2, and D3 are riddled with errors and stemming reduces precision significantly. Errors are because of ambiguous names and poor and no stemming. Pattern Matching is the most useful approach in isolation when name indicators are present. Newswire catalog lookup provides most gain.

5.2.4.2 Statistical based approach

Conditional Random Fields results show variations of results in each fold of 5-fold cross-validation, where 80% of the data was used as a training and 20% as a test. Average F-measure is 75%, with a 4.82 standard deviation. The error analysis showed that most errors were because of incorrect annotations, transliteration errors because of the mixing of Latin and Cyrillic characters, and over cleaning and removing of punctuations.

5.2.5 Analysis

Although this is one of the first works of Russian NER baseline, the corpus provided is quite limited and small. The corpus has 1% Person names and 2% organizations. Such statistics are quite unusual in a news feed; some analysis shows that choices made by annotator to ignore some names, and organizations contributed to this small number. The authors acknowledged that the small corpus size and limited tagging coverage has overtrained the statistical model.

The knowledge-based approach is a combination of pattern matching and dictionary lookup. The dictionary lookups only reported low performance because of input text and dictionary entry mismatch. Besides the mismatch, the dictionary entries were too noisy. A similar pattern was observed with Persian where the manual creation of gazetteers or dictionaries is error prone due to the lack of governance and results in poor performance. Although there are no similar resources available for Urdu like government registries that were used to create D1, I attempted to create a lookup based on DBPedia infoboxes like D2 and D3 dictionaries. The evaluation of entries in Urdu DBPedia and Infobox show that they are ripe with errors and anecdotal information. There is no section where named entities like organization can be found. There is no catalog in Arabic-Script for company or people names available from the news organization like BBC Urdu, Voice of America or any Pakistani newspapers that I contacted. Pattern matching is a subcomponent of the knowledge-based system in Gareev's s pipeline. These patterns are hard coded and look at prefixes only to determine the type of the entity. The patterns don't use linguistic information to determine the end of the noun phrase and hence run into false positives.

The results show that stemming of tokens reduces the performance significantly during a lookup because of stemming conflated names to the common words and nouns. Although the capitalization should be able to lift the F1 measure, this did not happen because many of the person names were not capitalized. Urdu suffered the same fate when stemming is used before named entity algorithm was run. The degradation in Urdu results was quite drastic because Urdu does not have determiners and

capitalization. Many times the anchor text and cues are also stemmed in Urdu. For example, *ministry* and *minister* will stem to *minist*. Consider the following sentences in:

S1: A divine is **most merciful**

Urdu translation of S1: خدا **رحیم** ہے

S2: Raheem ate an apple

Urdu translation of S2: **رحیم** نے سیب کھایا

In the above example, *most merciful* translates رحیم pronounced as *raheem. Raheem* is also a person and is written and pronounced exactly as **رحیم.** If a stemmer is used on Urdu sentence it will conflate to رحم *(Rahm)*. رحم is an Urdu word with ancestry from Arabic meaning *merciful* and a womb depending on the context, but it is not a person name. Therefore, Urdu representation of S1 after stemming خدا **رحم** ہے is a valid and good sentence that only changed the original the superlative form of adjective to positive form. In contrast, Urdu representation of S2 after stemming **رحم** نے سیب کھایا changed the subject completely from a person name to a non-person name, and therefore created a non-recognizable name. The above examples show that the effect of stemming on Urdu NER could be very drastic.

5.3 Arabic

Arabic is considered one of the six major languages of the world. Although many Indo-Iranian languages use the Arabic script, Arabic belongs to the Semitic group of languages which also includes Hebrew, Aramaic, and Amharic (in Ethiopia). Arabic is written from right to left like Persian and Urdu. Arabic is a highly inflected language with a complex syntax and a rich morphology. Arabic is considered an MRL with a complex morphology.

Arabic is the language of the Quran and is widely used in the Muslim world. It is obligatory for Muslims to be able to able to read it and recite it. Most of the Muslim population does so without complete understanding. There are many dialects of Arabic.

- Classical Arabic is the language of the Quran which was mostly spoken in Makkah, the birthplace of Prophet Muhammad.

- Modern Standard Arabic is used in newspapers, television, and official communication among Arab delegates and well-educated individual in conferences.
- Local dialects are found all over the Arab world where an individual from Egypt may not be able to understand someone from Morocco or someone from a Bedouin tribe in Saudi Arabia.

5.3.1 Introduction

Arabic is not a resource-scarce language because there is significant work available for Natural Language Processing. Khaled (2014) presents a comprehensive review of Arabic NER. Arabic has one of the most complex naming systems. Arabic naming system has had a significant influence on the development of proper names in the Islamic world. The names that are used by Christian Arabs follow a western based naming system, but this is not always true. Arabic naming systems continue to evolve. An Arabic name could have up to five different constituents, the Ism, the Kunya, the Nasab, the Laqab, and the Nisba.

- **Ism** -- this is like the Western given name, reserved for children and young people. It is frowned upon to refer adults, even family members with this name.
- **Kunya** -- Also known as Kunyat in Urdu and Farsi. Adults may be addressed using their Kunya. The refers to the person as a father or a mother of their oldest child. For example, Abu Ali *(Father of Ali)* or Um Hassan *(Mother of Hassan).* If a person does not have a child, an arbitrary noun or some other arbitrary names is the place.
- **Nasab** -- Also known as Nasb in Urdu and Farsi. This constituent is sometimes similar to European patronymic names. These names are used in Arabic as ابن ، بن *(Bin or Ibne)* for son and بنت *(Binte)* for daughter. For example, Usama Bin Laden
- **Laqab** -- A number of Arabs are referred by a laqab, that is a positive characteristic, and this is the often the preferred name. A common laqab all over the Islamic world is *Abd al* meaning *Servant of*. This is a prefix of many names that follow many possible names of Allah. For example, *Abd al-Hakeem, Adb al-Maroof* are examples of laqab. However, *Adb al* is a very common construction can be used as Ism also in many Arab and non-Arab areas. A concrete example of laqab is Abu Turab *(Father of*

Earth). One of the Islamic historical figures is referred as both Abu al-Hasan and Abu Turab. Where al-Hasan is Kunya, and Abu Turab is Laqab. Islamic literature is full of referring to this individual by either of those names.

- **Nisba** -- Also known as Nisbat in Urdu and Farsi. It is usually the description of the place of origin of the family; sometimes it can be the description of the occupation. This phenomenon is similar to the West (e.g. Goldsmith) or Farsi (e.g. Qumi (from Qom)).

Arabic names also have honorific titles also like شیخ *(Sheikh, Shaykh, Shaikh, Cheikh)* or حاجی *(Haji, Hadji, Hagi).* Sheikh is usually referred to a learned scholar, and Haji is referred as someone who has performed pilgrimage to Makkah. There are female versions of these names also.

Arabic script processing has many challenges some of which are shared by Persian and Urdu. Some examples that Zaghouani (2012) et al. mentioned are:

- Lack of diacritics in Arabic text that cause ambiguity in meaning and can only be solved by contextual information.
- Arabic has no capitalization, and hence there is no indication of the beginning and ending of a named entity like available in Latin script languages. Arabic has a determiner that can help identify a proper noun
- Arabic morphology is probably the most complex of all Arabic script languages. It is based on the root-pattern schema and has high inflection.
- Like any Arabic script language, there is lack of standardization of Arabic spelling for non-Arabic words or names. For example, حمید کرزائ can be written as Hamid Karzai (incorrect) instead of Hameed Karzai.

5.3.2 Approach

Shalaan's (2014) review of Arabic NER approaches concludes that because of the complexity of Arabic names there is a constant need for gazetteers. The MRL property of Arabic is challenging for Machine Translation and Transliteration. The machine learning based systems have low accuracy because of the quality of the available linguistic

resources despite their abundance. The rule-based approach tends to perform most optimally. In this section, the most effective approach used for Arabic NER by Zaghouani et al. (2010) is evaluated since it is considered the most effective NER system (Shalaan 2014). Zaghouani et al. (2010) is most similar to the Urdu NER system as presented in this dissertation with the exception of gazetteers lookup since they are not available for Urdu.

Zaghouani et al. introduced Arabic NER system (RENAR) while adapting a multilingual system European Media Monitor (EMM) for Information Retrieval and Information Extraction based application News Explorer for Arabic. The EMM system is optimized for rule-based language processing. RENAR used handwritten rules that are language independent and enhances them with Arabic specific rules and resources like dictionaries and gazetteers, and local grammars.

The following components were enhanced in the EMM system for Arabic adaptation

- Trigger words: Proper names are usually surrounded by cue words such as titles or verbs like *declared* or *said.* A list of 3,400 trigger words was introduced including 1,100 modifiers.
- Stop words: A list of 1,009 stop words from the Arabic corpus were identified as non-proper nouns.
- Gazetteers: A name gazetteer of 19,600 of names was introduced which included 17,000 first names with Arabic and non-Arabic origin, e.g., John. A number of name categories like Kunya, Nasba, etc. were introduced like *Abu*, *Abd*, *Ben*. The rest of the entries are last names. A simple gazetteer enhanced by Arabic Wikipedia is used for location. There are no rules created for location; all location identification is done by gazetteer lookups. The organization list contains 4,000 companies, the organization names that we extracted from the ArabiCorpus. Only the major organizations in the Arab world were chosen because of the complexity and the variations of the organization names.

- A preprocessing step was created for the morphological based language processing. For example, the conjunctions و *(wa)*, preposition ل *(li),* and determiner ال *(al)* are stripped for better match in the dictionary.
- The hand-crafted rules are regular expressions and are stored in language specific component of the overall system. The rules are fired based on trigger words and the if a name is not in a known dictionary identify

5.3.3 Corpus

Arabic is not a resource-scarce language regarding corpora in its native script. RENAR claimed that they used freely available Arabic Corpus (AC). There is no need to have an annotated corpus for RENAR because it is a rule-based and a dictionary lookup system. The Arabic Corpus is a collection of about 69 million words from various sources such as newspapers, Quran, and Arabic literature. There is a UI available to explore the corpus. This corpus is by far the most comprehensive and versatile corpus that was encountered in this research.

Although the corpus is available freely for online browsing and searching, it is not available for download or through API per inquiries. There is no mechanism to search without a query. For example, we cannot ask to display all nouns in the Islamic Discourse section of the corpus (a corpus in AC). Once a query is sent the corpus statistic show summary, citation, collocates, subsections, words before/after, and word forms.

5.3.4 Evaluation

RENAR did not use the entire AC corpus for their evaluation. Zaghouani et al. (2010) used 316 articles and 150,286 tokens which were collected from various online news sources. Proper names represented 11% of the news corpus, and their ratio of distribution within Proper names is shown below:

Category	**Ratio**
Person	39%
Location	30.40%

Organization	20.60%
Miscellaneous	10%

Table 18

The distribution above suggests that RENAR rule-based system with lookup is used only on about 3% of the experimental corpus that represented Person names. The location and organizations are dictionary looks ups. RENAR results are shown below:

Category	F-measure	Technique
Person	61.54%	rules + dictionary
Organization	52.23%	dictionary
Location	87.63%	dictionary
Overall	67.13%	dictionary

Table 19

RENAR compared their results with Lingpipe Arabic NER system and showed that their results are comparable to state of the art for Person and Organization categories and exceeds in the location category. All versions of Arabic names are found it Urdu text

5.3.5 Analysis

It is quite clear that Arabic is not a resource-scarce language, but the machine learning annotated resources lack accuracy and are very noisy to have a statistically-based robust system. NER rule-based systems are best of for complex language like Arabic (Shalaan, 2014). Zaghouani et al. (2010) described major challenge in RENAR was to create rules because of the complexity of Arabic naming system. Many times Arabic names in newspapers could be up to eight or more tokens, and therefore, boundary detection can become quite challenging. Moreover, Arabic names especially organizations are composed of regional variants. Therefore, local grammar rules could not account for all variations. For example, the Western Arab countries known as Maghreb has French influence in names such as – *Mohamed Daoud* ***Pieces*** *Auto*, and the Gulf region has English influence such as *Madani* ***Tailors***. Good performance of the location category is

explained by the good quality of the gazetteers which were created from the Wikipedia page. The low performance of the organization is because of the inconsistencies of the transcription system of the foreign companies to Arabic, long names, and low coverage of the names in the organization gazetteer. The majority of the person names problems are because of the transcription errors and inconsistent use of Arabic keyword.

Urdu NER system is most similar to RENAR in terms of the rule creation. The Becker-Riaz corpus contains a significant number of Arabic names, and some rules address infixes like *Bin,* and prefixes like *Sheikh.* NER system works with the 6-token window, and therefore it will not be able to recognize long Arabic names. Arab names have an advantage that it has an elaborate naming scheme over Urdu. Since Urdu does not have a determiner, it cannot determine the presence of a proper noun like Arabic proper noun indicator *al*. RENAR's preprocessing step removes the determiner to match the dictionary entries which reduces its recall. The boundary detection in RENAR is done by stop words, in contrast to Urdu NER system that uses grammatical cues, postposition, and case markers to determine the end of markers. The Urdu NER system does not use stop words because the approach of Urdu words detection sometimes recognizes proper nouns because they are found with high frequency in the corpus. (Riaz 2007)

5.4 Summary

An analysis of pioneering and known NER systems is presented. This analysis covered morphological rich languages like Persian, Arabic, and Russian and contrasted their approaches with the approach of Urdu NER system. Persian and Russian are considered resource-scarce languages. Despite the availability of resources, the quality of resources is of utmost importance. The creation of annotated resources for statistical learning model requires considerable manual effort, linguistic experience, and supervision. The cost to engage such resources is cost prohibitive in most situations. The most effective way to build an NER system for morphological rich language is rule-based because the effort and governance required for creating an annotated training set which is error free and has enough coverage can cost prohibitive.

6 Utility of NER on Search

One of the goals of this research is to show to that one of the utility of NER is to aid in the **about-ness** of the search. This section demonstrates that indeed NER can help in improving the relevancy of the results as they are presented to the user. NER enhanced search is most helpful when languages being queried are MRL. Moreover, it is shown that transliteration is Achilles' heel for name matching purposes (Lisbach 2015).

Experiments are designed for Russian, Arabic, English, and Urdu. The choice of languages is because of the availability of content, research system and enabling technologies. It is assumed for these experiments that an NER system exists for these languages and documents are indexed by them, although this assumption is not valid in all situations. As experiments are described, it will be clear the state of each system, i.e., if stop words and stemming has been applied to the index and the ability to search within metadata.

Evaluation

The evaluation measure for experiments is standard recall R, precision P, and F1-measure (F_1) as describe in chapter 3. In addition, we introduce Precision at rank n (P_n), which denotes how many relevant documents are present at rank$n$.

We provide the formulas again for clarity:

- $R = \frac{|relevant \cap retrieved|}{|relevant|}$
- $P = \frac{|relevant \cap retrieved|}{|retrieved|}$
- $F_1 = \frac{2PR}{P+R}$

These quantities can be explained concretely by the following example in a keyword-based search engine test bed. In this test example documents have the synonymous terms of *cars* and *automobiles*. Let d_1, d_2, d_3, d_4 and d_5 be a collection of five documents and a user executes a query q

$d_1 = \{car, red, fast, insuracne\}$

$d_2 = \{automobile, coverage\}$

$d_3 = \{automobile, color, fast, premium\}$

$d_4 = \{cars, classic, liability\}$

$d_5 = \{autobahn, Germany\}$

$q = \{car\}$

Result set of query q is $\{d_1, d_4\}$ when stemming is performed on the words in the collection and only $\{d_1\}$ without stemming. There are 4 relevant documents in the collection.

$R_{stemmed} = \frac{2}{4} = 0.5 = 50\%$ $R_{\not\ni stemmed} = ¼ = 0.25 = 25\%$

$P_{stemmed} = \frac{2}{2} = 1 = 100\%$ $P_{\not\ni stemmed} = \frac{1}{1} = 1 = 100\%$

$F_{1\ stemmed} = \frac{2(0.5)(1)}{1+.5} = 0.66 = 66\%$ $F_{1\not\ni stemmed} = \frac{2(0.25)(1)}{0.25+1} = 0.40 = 40\%$

The extent and rigor of the evaluation for each of the chosen language is based on the richness of the corpus, and the expertise needed to evaluate the results. For example, as we are not familiar with the Russian language we utilized the Google Translate to examine the named entities the Cyrillic script queries, and evaluation of native Cyrillic document returned by the Russian IR system. We are familiar with Arabic but not at native level, so we have a more targeted evaluation than Russian. The English evaluation shows many issues a search engine can encounter because of polysemy and transliteration. The most detailed evaluation is done for Urdu since it encompasses the most polysemy and homonymy features in Arabic Script languages. Moreover, Urdu is a morphologically rich language that has scarce linguistic resources which we remedied by providing such resources as part of the research contribution

6.1 Evaluation of utility of NER for Russian

Russian is considered a resource-scarce morphological rich language for NER systems. The analysis showed that NER system evaluation and baseline was done on about 100 documents and the resources used were low quality. This section will show that if a

Russian Cyrillic script based search engine was boosted with named entities, it would improve the **about-ness** of the documents retrieved.

A TF-IDF Lucene-like search engine is used for Russian evaluation which has large stores of Reuters News content in Russian Cyrillic script. The search engine supports Russian stemming and supports search in native Cyrillic script. Russian is not a right to left language so no special character and encoding issues were encountered.

The Russian language corpus is created by Reuter's newsroom in Russia. Some documents are directly authored by the Reuters News Bureau as they file stories, and some documents are an aggregation of stories of the day with a disclaimer. There is a classification engine runs on the corpus to tag with *subjects, location, regions,* and people. These tags are not verified by a human annotator to see the accuracy. The Russian corpus has 752,685 documents. The engine has the ability to restrict the corpus by date. NER-enhanced engine is evaluated on the document collection of 24,335 documents from the year 2107.

6.1.1 Evaluation in Russian

For the query selection process, 15 different queries were constructed in the Cyrillic script that included names of people, locations, and numbers. The queries included verbs or actions that provide context to the query. The process of selecting the query is quite tedious since I am not familiar with the script. A browsing system can be used to select documents. The top displayed documents titles were picked and translated using Google Translate from Russian to English. The document titles mentioned news stories in the Balkans region, former Soviet states and other common themed global issues such as Israel and Palestine conflict. Queries were selected that showed polysemy in query terms. Some queries were selected could use stemming. For example, Chechen and Chechnya could stem to a common stem.

6.1.2 Relevance Ranking based on pooling

There is no manually created relevance judgments for the Russian corpus because of its massive size. Therefore, pooling technique is used on the Russian engine. Multiple ranking

techniques are explored to evaluate the top 25 ranked documents to determine the relevance ranking of a query. The common documents from each method were considered to be relevant.

6.1.3 Experiment and Analysis

Russian evaluation consisted of running 15 queries for P_3, P_5 and P_{10} to input into the F_1-measure for baseline. Same measures were evaluated when boosting the query with Named Entities (NE). Five example queries are used for illustration and for deeper analysis.

- $\boldsymbol{q_1} = \{\text{подтвердило рейтинг Украины }\}$

/подтвердило/ *confirmed*

/Рейтинг/*rating*

/Украины /*Ukraine/* is NE

The concept that the user is seeking is **about** *the credit rating for Ukraine*

$q_1$retrieved 1,677 documents from 2017 date restricted corpus. There are 17 relevant documents to the query. Stemming was applied to all of the results. Evaluation of **about-ness** is done by examining the metadata and extracting the *metadata.preview* field and sending it to Google Translate and evaluating if the document was on point. At times *metadata.preview* field in the metadata was too small and did not have enough information. The engine has the capability to highlight the *hit* token to determine if the hit was on stemmed term or the exact term. This is a tedious and time consuming process and depends significantly about the accuracy of Google Translate. The terms in q_1 are treated equal and because of stemming Ukraine and Ukrainians are conflated together. For example, the baseline results saw a number of documents about Ukrainian language and Russian language in Crimea.

Documents in the Russian news corpus are tagged with the location information at least for the relevant documents. $q_1$to is enhanced by the location tag of *'Ukraine'.* It is important to note that the document metadata is in English and all value-added tags are

in English. In addition, boosting is applied on the Cyrillic script term Украины meaning Ukraine. The enhanced query is:

- $\boldsymbol{q_{1,ne}} = \{\text{подтвердило рейтинг Украины} < 2 > \ \& \ location(Ukraine)\}$

This query is executed as a hybrid Boolean and TF-IDF Ranked Retrieval query, and results are evaluated for P_3, P_5, and P_{10}

The results are tabulated below[18]

	q_1	$q_{1,NE}$
R_{25}	$\frac{8}{17} = 0.47$	$\frac{13}{17} = 0.76$
P_3	$\frac{1}{3} = 0.33$	$\frac{3}{3} = 1$
P_5	$\frac{1}{5} = 0.20$	$\frac{4}{5} = 0.8$
P_{10}	$\frac{3}{10} = 0.30$	$\frac{8}{10} = 0.8$
$F_{1@10}$	$\frac{2*0.30*0.47}{0.47+.30} = 0.366$	$\frac{2*0.8*0.76}{0.76+0.8} = .78$

Table 20

- $\boldsymbol{q_2} = \{\text{экспортирует продовольствие Россия Чечня}\ \boldsymbol{\}}$

Экспортирует /exports/

Продовольствие /food/

Россия /Russian/

Чечня /Chechnya/

Translation: Russia food exports to Chechnya

q_2 is **about** *Russia exporting food to Chechnya.*

[18] We show the breakdown of measures in detail for q_1, we will discuss the results of other queries inline.

The pooling techniques showed that there are only 2 relevant documents to $q_2$in 2017 date restricted collection. When $q_2$was executed as an OR Boolean query 2,076 documents were retrieved and none of the top 10 document were relevant. Therefore, $P_{10} = 0$. The query was modified from q_1 to boost to the location named entity by adding the *location* metadata to the query. Unfortunately this only helps in the better performance at P_{10}. Error analysis is explained below:

- A large number of the documents are already tagged as *location (Russia),* and the location tag is only available for countries and not for cities or provinces, hence no location tag for Chechnya.
- The fact that the terms *Russia* and *Russian* are stemmed as *Russia.* The term *Russia* has a low IDF in the corpus that it is acting like stop words despite the classification tag.
- Chechnya and Chechen were stemmed were stemmed together, and many documents were about the Chechen freedom fighters and their potential link to Al-Qaeda.
- There are two acceptable tokens for Chechnya as Чечня and Чечню.

q_2 was modified to boost Chechnya by 5 and restricted the query to match exact terms only i.e. Russia is restricted to match as is with no conflation.

$\boldsymbol{q_{2,NE}} = \{\text{экспортирует продовольствие } \& \text{"Россия" } \& \text{"Чечня"} < 5 >\}$ the results are much more encouraging $P_3 = \frac{1}{3}$, $P_5 = \frac{2}{5}$, $P_{10} = \frac{2}{10}$. The Recall is $R = \frac{2}{2} = 1$ and $F_1 = 0.36$

- $\boldsymbol{q_3} = \{\text{Казахстан в октябре мазута через каспийский порт Курык}\}$

Казахстан */Kazakhstan/NE*

октябре */October/*

мазута */fuel oil/*

через */via/*

каспийский порт Курык */The Caspian port of Kuryk/NE*

Translation*: Kazakhstan in October fuel oil via the Caspian port of Kuryk*

The query is **about** the country of Kazakhstan and the movement of oil or fuel via the Caspian port of Kuryk.

The total number of relevant documents are 19 for the top 25 ranked documents. Execution of $q_3$using a Boolean-OR algorithm and restricted date brought back documents 6,378 documents where most of the top documents were non-relevant. Boolean-AND algorithm returned 24 documents but most of them were non-relevant. $P_1 = 0, P = \frac{1}{5} = 0\ and\ P_{10} = \frac{2}{10}$. Analysis showed that since Russian language has high flection, number of documents returned were about Kazakhastan***i*** people. Some documents were about the importance of the Caspian port of Kuryk. A number of top hit documents were the aggregated news articles where one story is about the Kazakhastani people, and the other section in the same story is about the Caspian port of Kuryk. Moreover, some documents returned were filed by reporters in October. The hits on the stop word в which means *in* in English shows that the engine does not have stop words enabled.

Since Kazakhstan is a country and the documents in the corpus are tagged with country names. The query is modified to not get a hit on stop words of в *(in)* and порт *(of)* and boosting of named entities in the Cyrillic script. The query is executed in the hybrid Boolean and TF-IDF Ranked retrieval mode.

- $\boldsymbol{q_{3,ne}} = \{location(Kazakhstan)\ and\ (\text{Каспийский}\ within\text{: 3 Курык}) < 5 >\}$

The modified query retrieved 17 documents with 4 non-relevant documents

$P_3 = 1, P_5 = .8, P_{10} = 0.7, R = .78$, and $F_1 = 0.74$

- $\boldsymbol{q_4} = \{\text{ Торговые сделки Китая и Пакистана}\}$

Торговые /Trade/

Сделки /deals or transactions/

Китая /China/NE

Пакистана /Pakistan/NE

Translation: China and Pakistan trade deals

The Russian language has synonymy for the term Сделки as either *transactions* or *deals.* There are about 13 relevant documents in the date restricted collection. Boolean-OR query returned 2,681 documents where each term as equal weights. The Boolean-AND query retrieved 58 documents where only 3 were relevant at Rank 25. Precision values are $P_3 = 0, P_5 = \frac{1}{5}, P_{10} = \frac{2}{10}$.

Since the query contains two countries, it was enhanced with the *location* tag executed as *location(Pakistan) & location(China)* and Сделки. This new query did not improve results because most location tags in the document were *Asia*. Both Pakistan and China are in Asian countries. Instead of using *Asia* as a location tag, query was modified by boosting the terms Китая*/China/* and Пакистана*/Pakistan/*

- $\boldsymbol{q_{4,ne}} = \{\text{Торговые сделки Китая} < \mathbf{5} > \text{and Пакистана} < \mathbf{5} >\}$

The modified query returned 23 documents with duplicates at rank 4 and 5, and at rank 6 and 7, they are marked as correct if relevant. The results show $P_3 = \frac{2}{3}, P_5 = \frac{4}{5}, P_{10} = \frac{8}{10}$. $P_{10}$is high because of duplicate relevant documents. $R = \frac{11}{25}$ and $F_1 = 0.56$.

- $\boldsymbol{q_5} = \{\text{Саудовская Аравия дипломатии в Израиле и Палестине}\}$

Саудовская Аравия /Saudi Arabia/NE

Дипломатии /diplomacy/

Израиле /Israel/NE

Палестине /Palestine/

Translation: *Saudi Arabia's diplomacy in Israel and Palestine*

Pooling method suggested q_5 has 15 relevant documents at Rank 25. This query has three named entities of location and all are countries. Boolean-OR method retrieved 3,167 documents and none of them relevant. A date restricted Ranked Retrieval query without boosting is used as a base-line. The base-line query showed $R = \frac{6}{15} = 0.4, P_1 = \frac{1}{3} = .33, P_5 = \frac{2}{5} = 0.4$ and $P_{10} = \frac{2}{10} = 0.2$. Numerous documents during analysis showed the location tag of Saudi Arabia and Israel. So the query was modified to add location tags.

$q_{5,ne,1} = \{location\ (Saudi\ Arabia) and\ (Palestine) and\ location\ (Israel) and$ дипломати $\}$

This returned 3 documents with the Recall of 0. The analysis showed that only one of the returned documents had the term Saudi Arabia in Cyrillic and was tagged with Saudi Arabia which was actually about Saudi Arabia. The two other documents talked about September the 11th attack and Syria respectively. The analysis showed that a number of documents were tagged incorrectly. Boosting the query by named entities only with metadata tags gives a new query:

- $\boldsymbol{q_{5,ne,2}} = \{($Саудовская Аравия$) < 5 >$ Израиле $< 2 >$ Палестине $< 5 > and$ дипломатии$\}$

The second named entity enhanced query resulted in 13 documents with $R = \frac{9}{15}, P_3 = 0.33, P_5 = 0.6, P_{10} = 0.70\ and\ F_1 = 0.65$

The following table shows the comparison of F_1 measures between NER enhanced system versus the base line in Russian.

Sample Queries	$\boldsymbol{F_1 measure}$
q_1	0.37
$q_{1,ne}$	0.78
q_2	0.00
$q_{2,ne}$	0.50
q_3	0.13
$q_{3,ne}$	0.74
q_4	0.28
$q_{4,ne}$	0.57
q_5	0.27
$q_{5,ne}$	0.65

Table 21

The results on sample queries show that NER significantly improve the F_1 measure of the IR system. Multiple methodologies were used for the system to detect the named entities. In general, if named entities are indexed and detected with significant accuracy then the performance of and IR system can be improved significantly. The results in Russian show that improvement is on average 34% better than baseline for sample queries. The analysis of other queries showed similar challenges as shown in the analysis.

6.2 Evaluation of utility of NER for Arabic

Arabic is not considered a resource-scarce language, but there is still a scarcity of a good NER system for Arabic. It is considered a morphologically rich language. This section shows that a search engine with Arabic documents when enhanced with named entities improves the **about-ness** of the results. A commercial search engine based on Lucene-like algorithm is used for evaluation. The search engine has an Arabic stemmer available. We don't know if the stop words in Arabic are removed from the index. Error analysis showed that a stop word list is not enabled in the system.

The Arabic corpus is the rules and regulations of the Qatar Central Bank. The evaluation queries are very targeted and mostly focus on locations and organizations. Evaluation in Arabic was time-consuming because named entities needed to be determined for judgment. This particular corpus was chosen because it had access to a parallel in English of the Qatar Central Bank legislations. The Arabic corpus size is 388 documents. A number of encoding issues related to Arabic script languages were encountered. In Russian, to boost a term like Обам, <2> was appended as a boosting syntax. This addition of <2> boosted the term Обам with twice the weight. Since Arabic is right to left language the character representation in Code Point of '<' is '>' and the Code Point for '>' is '<'. This was discovered this after continuously observing no changes in ranking or weights of the retrieved documents after boosting. Although the number of queries for deep analysis are small, the analysis is quite detailed in how the synonymy, polysemy and orthographic issues while researching in a Arabic Script and morphologically rich language.

6.2.1 Relevance Ranking based on pooling

There are no manually created relevance judgments for the Arabic corpus. Pooling technique is used as was for Russian analysis to determine relevant documents and evaluation

6.2.2 Experiment and Analysis

Analysis and discussion is based on the following queries

- $q_1 = \{$المالية الأوراق$\}$

المالية /Finance/(wealth)

الأوراق /Bills/(paper)

Translation: *Money Bills*

This query is considered to be a proper noun in the Qatar Central Bank corpus given the domain. The phrase or word is constructed from two words:

- الأوراق which is the plural of ورق meaning paper
- المالية which means Finance. Since Arabic is a highly inflected language, there are many different derivations that occur from the root form of المالية. For example, مال، /maal/, المال/al-maal/ and a number of tense forms e.g. as it is used in مالى/maal-i/
- The combined phrase can also be interpreted as Financial Securities and therefore will introduce polysemy.

The query represents the named entity as one unit and the relevant documents will be **about** the complete phrase. q_1 has 9 relevant

When q_1 was run without boosting the query for named entities with Boolean-AND, it resulted in the retrieval set of 71 documents. The analysis showed that a number of top ranked documents were not relevant because they contained morphological variants of one of the phrase's terms. $P_3 = 0, P_5 = 0, P_{10} = 0.1$ The Recall at 25 was $R = \frac{3}{9} = 0.33$ and $F_1 = 0.15$.

For the analysis of this section, it is assumed that a high performant NER Arabic algorithm exists and it has tagged the documents. For the Arabic system's Lucene-like fielded search

is available in the original script. The query was modified to do a fielded search with named entity boosting, and stemming turned off. A hybrid Boolean and Ranked Retrieval model based on TF-IDF was executed.

- $\boldsymbol{q_{1,ne}} = \{Title(\text{المالية الأوراق}) \text{ المالية} < \mathbf{5} > \text{الأوراق} < \mathbf{5} >\}$

This query returned 17 documents, evaluation showed $P_3 = 1, P_5 = 1, P_{10} = .80$

$R = 1 \ and \ F_1 = 0.88$. This is a significant jump from the baseline attributed to the to these factors:

- The stemmer was disabled because the query was a named entity, and there were no verbs or common nouns in the query.
- The fielded search mimics a situation where all the named entities are extracted and fielded and searched as is without stemming

- $\boldsymbol{q_2}$ = {فرقة العمل المعنية بالجرائم المالية}

The phrase الجرائم المالية in the above query has been transposed because there is another construction within a larger phrase.

الجرائم المالية /Financial Crime/

الجرائم /Crime/

المالية /Financial/ *note how the same word changed from Finance to Financial*

فرقة العمل/teamwork/

فرقة/band

العمل /work/

المعنية /Concerned/

Translation: *Financial Crime Task Force*

This query shows the morphological complexity i.e. synonym, polysemy and derivation in Arabic. We consider this story is **about** the Financial Crimes and a Task Force that deals with them. The breakdown of the tokens clearly shows Arabic's complex morphology That

is, an individual token individually has one meaning and as they are combined in a phrase has a different meaning.

Pooling strategy showed that relevant documents for this query are 7. q_2, when executed as a Boolean-OR, resulted in 154 documents. It is almost half the corpus and most of the non-relevant results are because of stemming. The exact string search retrieved 0 results. The Boolean-AND query returned 4 document with only one being relevant. Most of the issues is missing the relevant documents is that stemming in conflating the proper nouns and there is no concept of phrase detection. $P_3 = \frac{1}{3}, P_5 = \frac{1}{4}$ and $P_{10} = \frac{1}{4}$, $F_1 measure = 0.18$

In an ideal NER system Arabic terms for Financial Crimes and Financial Crimes unit will be tagged as a named entity field and indexed. In the absence of these fields, boosting is used on the named entity tokens along with searching in the designated fields. The modified query below is run in a hybrid Boolean and TF-IDF format

- $\boldsymbol{q_{2,ne}} = \{title(\text{العمل فرقة}) \ title\ (\text{المالية بالجرائم}) \text{المعنية} (\text{المالية بالجرائم}) < 5 > (\text{العمل فرقة}) < 5 >\}$

The named entity enhanced query retrieve 13 documents 5 of them being relevant. $P_3 = \frac{2}{3}, P_5 = \frac{3}{5}, and\ P_{10} = \frac{5}{10}. R = \frac{5}{7} and\ F_1 - measure\ = 0.58$

Analysis of results showed that a stemming is hurting the recall also if there is cut off in place like Rank of 25 in the evaluation.

- $\boldsymbol{q_3}$ = { قطر }

قطر /Qatar/

On the surface the above query looks quite simplistic and it is **about** the country of Qatar. The Arabic morphological richness and its word and phrase creation cause a significant number of false hits. Moreover, an aggressive stemmer for this system retrieves a number of non-relevant documents. q_3 retrieved 109 documents, some of the terms that were hit in the top documents are provided below. The portion of the word that was hit on the search is bolded.

- ريال **قطري** – Qatari Ryal
- ورقم الهوية **للقطريين** والمقيمين: ID number for Qataris and residents

- مركز **قطر** –Qatar center
- دولة **قطر** – the State of Qatar
- ال**قطر**ية اليومية – Qatari Daily

There are 12 relevant documents for $q_3$about the state of Qatar.

The results of $q_3$show that $P_3 = 0, P_5 = 0, P_{10} = \frac{1}{10}, R = 0.16, F_1 - measure = 0.12$

In order to boost the presence of named entity, the query is enhanced with the fielded search as:

- $\boldsymbol{q_{3,ne}} = \{title(\text{قطر})\}$

Enhanced query retrieved all the relevant document but there was not as much lift in the precision. $P_3 = 0, P_5 = \frac{1}{5}, P_{10} = \frac{3}{10}, R = 1, F_1 - measure = 0.46$. The analysis showed that قطر and its morphological variants specially the compound words are hurting the performance. For example, most of the top documents per for the query were about the Bank of Qatar, which is not surprising because the corpus is about the Qatar Central Bank legislation. If a named entity was captured and indexed representing the state of Qatar it will help in improving the precision.

The following table shows the comparison of F_1 measures between NER enhanced system versus the base line in Arabic.

Sample Queries	$\boldsymbol{F_1 measure}$
q_1	0.15
$q_{1,ne}$	0.88
q_2	0.18
$q_{2,ne}$	0.58
q_3	0.12
$q_{3,ne}$	0.46

Table 22

We ran the experiments on 7 queries total 3 of which are represented here for discussion with the most analysis. It is quite difficult to find named entities because the nature of the corpus. The analysis was not as detailed on the 4 other queries but the general pattern of over stemming, and the morphological challenges were constant. The analysis of queries shows that there is a lift of 39% on average when a query is boosted with a named entity and stemming is restricted on the named entity.

6.3 Evaluation of utility of NER for Urdu

Urdu is considered a resource-scarce morphological rich language and the focus of this dissertation. This section shows that if Arabic Script Urdu language-based search engine was boosted with named entities it will improve the **about-ness** of the documents retrieved. There is no known Urdu search engine whose inner workings can be evaluated, or where the linguistic resources like a stop word list or a stemmer can be added. One of the major contributions of this research is to build such a system that does that. Architecture of the system shown in the figure below:

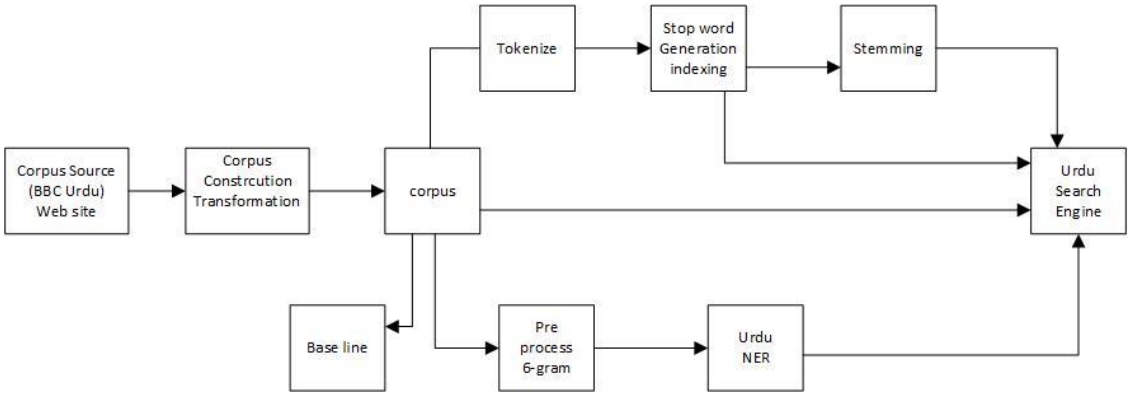


Figure 7

6.3.1 Evaluation in Urdu

For Urdu evaluation Becker-Riaz corpus (Riaz 2002) is used. As the architecture indicates, stemming, and stop word list is fed separately to the search index. This is done because stemming of named entities will result in low precision. We have used a number of

rudimentary to well-developed search engines. These engines ranged from home grown to open source Lucene to create the base-line (Riaz 2008). For this evaluation, Elasticsearch is used which is based on Lucene. Elasticsearch is chosen because it is a modern search engine and it has a capability of boosting terms. In addition, it can support Boolean query and TF-IDF query for baseline.

6.3.2 Relevance judgements

Urdu evaluation process and the selection of base line is much more thorough and rigorous than Arabic and Russian. A set of 200 relevance judgments that have been created (and evaluated by native speakers (Riaz 2008)). For each query the total number of documents that should be retrieved are known. For each relevance judgments a set of sub queries that can be created from original relevance judgment. For example, if a query was about *Daniel Pearl Case* there should be 12 documents that should be retrieved, their document numbers. In addition, relevant documents about *Daniel Pearl kidnapping, Daniel Pearl murder, possible suspects, Daniel Pearls wife etc.* are also identified.

6.3.3 Experiment and Analysis

Urdu evaluation consisted for 50 queries. Seven queries are showcased for the discussion and analysis in this section. The queries are chosen to represent the challenges of Urdu computational processing. Elasticsearch with boosting is used as a mechanism to evaluate Urdu Search so as much like-to-like comparison can be done with the results with other language evaluations.

The query selection of Urdu is deliberate and selective. Query are chosen where there is Polysemy in names, locations, and where stemming can introduce Polysemy. It is important to note that the behavior of Elasticsearch in the evaluation. That is, when a term is boosted at the query time, the most relevant documents will be ranked first. The number of documents retrieved before and after the boosting don't change, only the ranking changes. The stemming is applied to the search but the boot is only provided to the un-stemmed named entity. For example, if the query was اسرائیل/$Israel$/, only the term اسرائیل is boosted not اسرائیلی /Israeli/.

- $\boldsymbol{q_1}$: {اسرائیل و فلسطین}

اسرائیل /Isreal/

فلسطین / Palestine/

و /connector

Translation: *Israel-Palestine*

This particular query is about the Israel and Palestine conflict and what makes it interesting is the connector between the two words. In English, such a connecter is used as a hyphen between the two words. This query has 22 relevant documents. The baseline query returned 119 documents. An interesting pattern is observed where the و was playing a critical role in bringing up documents that had phrases which were constructed by this Persian based compounding for words to form another word. For example, the top documents had phrases like خوشحالی و آزادی /happiness/ and حرکت و نقل /movement/ اقارب و زیز/family and friends/. This is because that و does not occur alone usually and if it does it is because of this Arabic and Persian construction of connecting two words for emphasis. Hence, it has a high IDF and decent Term Frequency to be ranked higher than other meaning bearing named entities. The evaluation measures for base line are provided below. The baseline results for$q_1$are $P_3 = \frac{1}{3}, P_5 = \frac{2}{5}, P_{10} = \frac{4}{10}, R = \frac{8}{22}, F_1 - measure = 0.38$.

Both Palestine and Israel are recognized by the Urdu NER system, so the query is enhanced by boosting the named entities. The new enhanced query with elastic boosting syntax is provided below

- $q_{1,ne} = \{$و اسرائیل^5 فلسطین^5$\}$

This query brought back 119 documents but the documents about Israel and Palestine were ranked much higher. The connecter و stilled played a significant part after P_{10} and thus reduced the recall. The results showed that $P_1 = \frac{2}{3}, P_5 = \frac{4}{5}, P_{10} = \frac{8}{10}, R = \frac{17}{22} and\ F_1 - measure = .78$. Some of the non-relevant documents are about Israel and Iran tension, Israeli elections and general Arab-Israel diplomacy.

- $\boldsymbol{q_2}$ = { بغاوت نیپال}

نیپال /Nepal/

بغاوت/Rebellion/

This query is **about** the rebels and rebellion in Nepal by the Maoist rebels. There is only one named entity in this query. The relevance judgment for this query suggests 13 relevant documents. The baseline query without any name boosting retrieved 18 documents, $P_3 = \frac{2}{3}, P_3 = \frac{3}{5}, P_{10} = \frac{3}{10}, R = \frac{5}{13}. F_1 - measure = 0.33$. The non-relevant queries were mostly about mountaineering adventure in Nepal and rebellion in Turkey and rebels in Manilla. Since Nepal is an identified named entity recognized by the Urdu NER system the query is modified to boost Nepal:

- $\boldsymbol{q_{2,ne}}$= { بغاوت نیپال^5}

The modified query returned 18 documents. $P_3 = \frac{3}{3}, P_5 = \frac{3}{5}, P_{10} = \frac{8}{10}, R = \frac{12}{13}, F_1 - measure = 0.85$.

The analysis showed that a number of document were about Maoists rebels ranked lower than non-relevant documents so the query was modified with the term Maoists. Since Maoists are treated as a proper noun in the corpus it will be treated as a named entity and boosted it. As a new query term is added, it will be considered new query.

- $\boldsymbol{q_3}$ = { بغاوت ماونواز^5 نیپال^5}

ماونواز /Maoists/

Results of the new query are no different than $q_{2,ne}$. This was surprising because an improvement was expected. Analysis showed a number of typographic errors or stylistic preferences of journalists regarding how to represent writing Maoists in Urdu. Examples are provided in the Urdu script and the issues representation in English:

a) ماؤنوازبغاوت */MosistRebellion/* -- no space between *Maoists* and *Rebellion*

b) ماؤ نواز باغیوں */Mao ist Rebellion/* -- *space between Mao and ist*. In addition, the representation of *ist* is written as *Nawaz*, which is proper name in the Urdu corpus

representing the first name of Pakistan's politician and of and on Prime Minister Nawaz Sharif.

c) ماؤنوازباغیوں/*MaoRebels -- no space between Mao ist and Rebellion.*

One way to solve the issue if it occurred in English is use Soundex and Levenshtein distance but the matching algorithm does not work on Arabic script languages (Round 2017). We modified the query and only introduced named entities which are boosted.

- $\boldsymbol{q_{3,ne}}$ ={ نیپال^5 *ماؤنواز^5}

Execution of this query results shows $P_3 = 1, P_5 = \frac{4}{5}, P_{10} = \frac{9}{10} and\ R = 1, and\ F_1 - measure = 0.95$. The analysis shows that the mentioned typographical errors or stylistic issues were all addressed. Although this kind of regular expression matching is not advisable in a user facing engine, it does enforce the hypothesis that named entity plays an important role in the **about-ness** of the query.

- $\boldsymbol{q_4}$={چیچن گروزنی روسی کاروائی}

چیچن/Chechen/

گروزنی/Grozny/

روسی/Russian/

کاروائی/Action/

Translation: *Russian action Chechen Grozny*

This query has only one named entity without stemming but three if stemming is done on Chechen → Chechnya and Russian → Russia. The goal in this query is to explore the effect of stemming on the query terms. When the baseline query was run with all terms at equal weights the following results were observed: $P_3 = 0, P_5 = \frac{1}{5}, P_{10} = \frac{1}{10}, R = 1, F_1 - measure = 0.33$. Results showed that significant number of non-relevant documents ranked higher because due to the occurrence of term کاروائی/Action/ and روسی/Russian/ and were about Afghanistan or American *action* in Iraq. A few documents mentioned Al-Qaeda's links with

Chechen and Uzbek fighters. The query was changed to boost one recognized named entity گروزنی/Grozny/ as:

- $\boldsymbol{q_{4,ne}}$ = {چیچن گروزنی^5 روسی کاروائی }

The results of this boosted query showed $P_3 = 1, P_5 = \frac{3}{5}, P_{10} = \frac{3}{10} and\ R = 1, F_1 - measure = 0.46$. When stemming was applied to Chechen → Chechnya and Russian → Russia $P_{10} = 0$ was observed. This was because the number of documents where the terms *Russia* and *action* co-occurred together to boost the document scored higher. Query analysis shows that stemming of terms in a query that are named entities negatively impacts the precision. This behavior was also observed during the evaluation of the utility NER on Russian language.

- $\boldsymbol{q_5}$ ={افغانسان امن فوج}

افغانسان/Afghanistan/

امن/peace/

فوج/army/

Translation: *Peace force in Afghanistan*

In this query امن فوج means *peace force* which is a phrase. Urdu phrases are not indexed because of high polysemy of Urdu. The query is **about** the multinational peace force in Afghanistan, a relevant document could mention NATO or British or American forces in Afghanistan. This query also intends to show the utility of stemming on terms that are not named entity terms. The query execution without boosting on named entities and without stemming returned 154 documents. The number of relevant documents for this query is 7. $P_3 = \frac{2}{3}, P_5 = \frac{3}{5}, P_{10} = \frac{3}{10}, R = \frac{3}{7}, F_1 - measure = 0.41$. *Peace force* is not considered a phrase and there are considerable non-relevant documents about peace but not about *Afghanistan* or a *force*. Afghanistan is a recognized named entity for the Urdu NER system so boosting is applied on it. This resulted in improving P_{10} by one point, and increased the recall by finding one more relevant documents, hence the new$F_1 - measure = 0.53$. Although this has shown considerable improvement, and as expected documents about

Afghanistan were ranked higher, relevant documents are still missing. The analysis shows that non-relevant documents are ranked higher because the lack of stemming. The plural form of فوج/army/ is افواج/armies/ which a Persian loan word in Urdu for armies and is being pluralized using Persian morphology. Some of the relevant documents did not get a hit because the document contained the plural form. Moreover, the peace forces are represented using the Arabic loan words عسكري*/Askari/(Soldier)*, Persian loan word فوجی*/Fauji/(Soldier),* Persian loan word سپاہی*/Siphahi/(Soldier).* The Urdu Stemmer (Riaz 2007) will stem افواج /forces/ to فوج and فوجی /soldier/ → فوج. Re-execution of query using after re-enabling stemming.

- $q_{5,ne}$ = {افغانستان^5 امن فوج }

The stemming enabled system brought back 275 documents which is not surprising since stemming was enabled. The results showed: $P_3 = \frac{2}{3}$, $P_5 = \frac{4}{5}$, $P_{10} = \frac{6}{10}$ and $R = .85$ and$F_1 measure = 0.70$. The results show that the recall, P_{10}, and F_1 score improved when stemming was used on the non-named entity terms and the named entity was boosted for weights.

- q_6 = {عمر شیخ}

عمر/Omer/(*age* or *name*)

شیخ/Sheikh/(scholar or name)

Translation: *Infamous kidnapper and murderer of Daniel Pearl.*

This query explores ambiguity in names. The query contains two tokens and they both have polysemy. The relevant documents should be **about** kidnap, and murder of Daniel Pearl, the life and history and of Omer Sheikh, the fundamental organizations etc. Either of the tokens can be a first name or a last name of a person. In the Becker-Riaz corpus, there are a number of such scenarios e.g. Mullah Omer – the leader of Taliban, Omer Sharif – the comedian, and Haseena Sheikh – a politician in Bangladesh to name a few. There are 12 documents that are relevant. The documents that only discuss Daniel Pearl are not considered relevant to this query e.g. a story about Daniel Pearl's wife is not relevant to this query. Execution of this query with boosting where each term is treated

individually retrieved 55 documents with the following results: $P_3 = 1, P_5 = 1, P_{10} = \frac{5}{10}, R = \frac{6}{12}, F_1 - measure = 0.5$ The analysis showed that the half the documents before P_{10} were about individuals other than Omer Sheikh whose first and last name are *Sheikh* or *Omer* and one document discussed age related article because عمر mean age in Urdu.

- $\boldsymbol{q_{6,ne}}$ = {(عمر شیخ)^5 }

The named entity enhanced query shows two aspects, booting and searching for the individual as one entity because Urdu NER has recognized both together as an entity. The enhanced query retrieved 9 documents with $P_3 = 1, P_5 = 1, P_9 = 1, R = \frac{9}{12}, F_1 - measure = 0.85$

The analysis showed that some news stories are authored referring *to Omer Sheik* as *Sheikh Omer* transposing the first and second names. Urdu NER system has also recognized the transposed named but as a separate entity. This is considered story filing error.

- $\boldsymbol{q_7}$ **=** {اغواء پرل قتل کیس }

اغواء/Kidnapping/

پرل/Pearl/(first name of Daniel Pearl)

قتل /murder/

کیس/case/code-switching from English

Translation: *Kidnapping Murder Case of Pearl*

This query is **about** the kidnapping and murder of the Wall Street Journal journalist Daniel Pearl in Pakistan and the arrest and legal proceedings. The term *case* is used from English as it is used frequently in common speech. Its equivalent term is مقدمہ /muqadma/(case) in Urdu which is derived from Arabic. The Turkish loan word دعوی /dava/(*case)* is also used for legal proceeding in the corpus. There are 18 relevant documents for this query. The relevant documents could include the information about Omer Sheikh who was involved in Pearl's kidnapping and murder. Execution of q_7 with all terms at equal weights retrieved 69 documents. Since Pearl is a high IDF term 7 of 18 relevant documents were retrieved in top 10 ranked documents. The non-relevant documents about other murder

cases, and kidnapping cases ranked higher than other relevant documents. The results are: $P_3 = 1, P_5 = 1, and\ P_{10} = \frac{8}{10}, R = 0.5, F_1 - measure = 0.61.$

The named entity پرل /Pearl/ was boosted in the query because Urdu NER has recognized it as a named. The Urdu NER system has recognized both *Daniel Pearl* and *Pearl* as entities. The new query is represented as:

- $q_{7,ne}$ = {اغواء قتل کیس پرل^5}

The enhanced query all but one relevant document past rank 25. The results from the name boosted query are: $P_1 = 1, P_5 = 1, P_{10} = 1, R = 0.94, F_1 - measure = 0.97.$ The table below shows the F_1 measure of baseline and enhanced queries.

Sample Queries	$F_1 measure$
q_1	0.38
$q_{1,ne}$	0.78
q_2	0.33
$q_{2,ne}$	0.85
q_3	0.85
$q_{3,ne}$	0.94
q_4	0.33
$q_{4,ne}$	0.46
q_5	0.41
$q_{5,ne,nostem}$	0.53
$q_{5,ne,stem}$	0.70
q_6	0.5
$q_{6,ne}$	0.85

q_7	0.65
$q_{7,ne}$	0.97

Table 23

Analysis shows that Urdu NER improves on average 24% lift state of the art NER IR systems. Also, stemming plays an important part search but it hurts named entities.

6.4 Evaluation of utility of NER for English

English is neither Morphological Rich Language nor resource scarce language. Nonetheless, majority of the research in NER is based on English as a baseline. In this section it is shown that if an English search engine is boosted with named entities, it will improve the **about-ness** of the documents retrieved.

A search engine that is based on the modified version of Lucene and has large stores of multitude of News content is used for evaluation. BBC's Monitoring Service in South Asia is chosen as a content set since it has a number of names and issues from South Asia. The search engine supports English stemming, equivalency lists, and stop words. This section also shows that transliteration of non-English and non-Western names in the absence of a standard transliteration scheme can hurt the performance of IR system in both recall and precision.

The BBC Monitoring Service corpus is an aggregation of news stories that discuss multiple topics or discuss a topic at length. There is a classification engine runs on the corpus to tag with *subjects, location, regions,* and *people*. These tags are not verified by a human annotator to see the accuracy. The BBC South Monitoring Service corpus has 279,713 documents.

6.4.1 Evaluation in English

For the query selection process, 10 different queries are constructed that included names of people, locations and numbers. In general, the emphasis is placed on people names, and country names. Some of the queries are **about** the same concept as the queries as in other evaluations.

6.4.2 Relevance Judgments

Instead of using the pooling method for English a cut off technique and Rank 25 is used to determine the relevance because of evaluator's ability to read English. Since the English corpus has hundreds of thousands of documents, the quality of the corpus is commercial grade and the engine we are using is commercial grade, results will have at least 25 relevant documents.

6.4.3 Experiment and Analysis

The evaluation used 10 queries to show the utility of NER on English. A subset of the queries is used to show case the importance of NER. The engine that we use does not natively support named entity index.

- $\boldsymbol{q_1} = \{Nawaz\ Sharif\ Panama\ papers\}$

This query is **about** the former Prime Minister of Pakistan and money laundering implication in the Panama Papers. The execution of the baseline query using Boolean-OR retrieved 14,492 documents where the top documents are about Mazar-e *Sharif* – a northern area of Afghanistan, Shahbaz Sharif – the brother of Nawaz Sharif, a few documents are about Nawaz but not about Panama. The query illustrates the complexity of names and representing them in English through transliteration. For example, Mazar-e-Sharif is connected in Urdu through Izafat – a Persian morphology for compounding and the authors did not follow the proper technique for transliteration. The metrics for baseline Boolean-OR show$F_1 = 0.$ A Boolean-AND query returned 2 documents 1 relevant. The hit of Panama was transcription error of representing PANAM flight 103 as Panama flight 103. $P_{10} = \frac{1}{2}, R = \frac{1}{25}, F_1 = .07.$

The query was modified by boosting the named entity "Nawaz Sharif" and "Panama". The name and location tag are not used because the quality of the tagger is observed to be suboptimal for the documents returned.

- $\boldsymbol{q_{1(n,e)}} = \{(Nawaz\ Sharif) < 5 >\ Panama < 5 >\ papers\}$

The boosted query retrieved 37 documents with the following metrics. $P_3 = \frac{2}{3}, P_5 = \frac{3}{5}, P_{10} = \frac{5}{10}\ R = \frac{9}{25}, F_1 - measure = 0.41.$ Analysis of the results showed that there were not as many

documents that mentioned Panama. Instead, the stories filed about the money laundering scandal used the term *corruption* or *hiding money* overseas. Nonetheless, boosting the query with named entity at least ranked documents about *Nawaz Sharif* higher than non-relevant documents.

- $\boldsymbol{q_2} = \{Iran\ nuclear\ enrichment\}$

This query is **about** the Iran nuclear program and the relevant documents could contain uranium enrichment, the diplomacy talks, and comments about the world leaders. Execution of baseline system query returned 9,221 documents. A number of top ranked documents were about the North Korea nuclear program, and India's nuclear partnership with United States. The baseline metrics are: $P_3 = 1, P_5 = \frac{3}{5} and\ P_{10} = \frac{5}{10}, R = 0.52, F_1 - measure = 0.50.$

There are two ways to enhance the query, one is to use the term boosting on Iran, and second to add tagged classification such as *location(Iran) & (nuclear enrichment).* Boosting method is used for this query.

- $\boldsymbol{q_{2,ne}} = \{Iran < 5 > \ nuclear\ enrichment\}$

The above query boosts the weight of the term *Iran* which is a named entity. The execution of the results returns the same number of documents but in a different rank order. $P_3 = 1, P_5 == 1, P_{10} = \frac{9}{10}, R = 0.68, F_1 - measure = 0.77$. The non-relevant documents found in first 25 ranked were about Iran but not about its nuclear program. Some non-relevant documents are aggregated news article that discussed North Korea's nuclear program and Iran's involvement in Syria. Boosting on named entity clearly shows improvement in precision and recall.

- $\boldsymbol{q_3} = \{Saudi\ Arabia\ involvement\ against\ rebels\ in\ Yemen\}$

This query is **about** the civil war in Yemen and Saudi Arabia supporting one faction. The relevant documents could talk about tension in between Saudi Arabia and Iran over Yemen, the diplomacy talks, and Iran's support of the Houthi rebels.

The baseline query execution with all terms at equal weights resulted in a number of non-relevant documents. The top ranked documents were about Bangladesh and Saudi Arab's cooperation, Arabian Peninsula, Lebanese leader's visit to Saudi Arabia, Al-Qaeda (most articles used of Al-Qaidah as a spelling). The Houthi rebels are written as with multiple spellings e.g. Houthi, Huthi, Al-Huthi to mention a few. We encountered the familiar issue as in q_1 where a number of articles were classified about an entity when the article was not about it. In this case, a significant number of documents were tagged to be **about** *Saudi Arabia* but there was no mention of it Saudi Arabia in the documents. A number of hits were on *rebellion* as it is treated with equal weight as the named entities and *rebel* and *rebellion* conflate to the same stem. The results showed $P_3 = 0, P_5 = \frac{1}{5}, P_{10} = \frac{4}{10}, R = 0.44, F_1 - measure = 0.41$

Since a number of relevant documents were **about** rebellion in locations other than the entities mentioned in the query, and the analysis showed that using the classification tag will reduce the precision. The query is booted with named entities as:

- $\boldsymbol{q_{3,ne}} = \{$ ("$Saudi\ Arabia$") $< 5 >\ involvement\ against\ rebels\ in\ Yemen < 5 >\}$

The modified query takes Saudi Arabia as a complete entity and will not hit on terms like *Arabia* or *Arabian*. It does have a potential to not bring in some relevant documents. Results show the following results: $P_3 = 1\ , P_5 = \frac{4}{5}, P_{10} = \frac{6}{10}, R = 0.64, F_1 - measure = 0.62.$ Analysis showed that emergence of another named entity *Pakistan* as a broker between Saudi Arabia and Iran during this conflict. In addition, the documents show a *discovery* where the role of Pakistan as it is allied with Saudi Arabia and its generals are commanding the 34 nation Saudi led force in Yemen. Therefore, we have shown the named entity boosted search has discovered the **about**-ness user intention.

- $\boldsymbol{q_4} = \{Chaudry\ Nisar\ Pakistan\ interior\ mininster\ and\ Imran\ Khan\ \ negotiation\}$

This query is about Pakistan's politicians in opposing parties and their negotiations with each other. This query was chosen to show the ambiguity of names and transliteration problems. Chaudry Nisar Ali Khan was the interior minister of Pakistan when Imran Khan a famous cricketer turned politician was in the opposition party. The two politicians are

supposed to be friends outside of the political arena. Notice that both entities have *Khan* -as their last name. The baseline non-named-entity aware query returned 8,629 documents. Most of the top documents talked about *Chaudry Nisar* and his talks with Taliban – which is sometimes spelled as *Taleban*. A number of hits were just on document that contained only *Pakistan*. The ambiguity in names caused most of the issues i.e., a number of hits are on *Asghar Khan* – a general in Pakistan's army, *Aseem Khan*, and *Dr. Khan*. Transliteration problems are impacting the performance as the first name of the interior minister is represented as *Chaudry, Chaudhary, Chaudary, Chowdr*y either in the same article and across articles. Some articles about the talks between *Imran Khan* and *Chaudry Nisar* were ranked past rank 10. $P_3 = 0, P_5 = \frac{1}{5}, P_{10} = \frac{2}{10}, R = \frac{7}{25}, F_1 = .23$

The enhanced query boosts the named entities and groups the respective named entities together.

- $\boldsymbol{q_{4,ne}}$**=** $\{(Chaudry\ Nisar) < 5 > Pakistan < 5 > interior\ mininster\ (Imran\ Khan) < 5 > \ negotiation\}$

The enhanced query retrieved a number of relevant documents with the following results. $P_3 = 1, P_5 = 1, P_{10} = \frac{7}{10}, R = \frac{18}{25}, F_1 = 0.71.$ The relevant documents were **about** the discussions between the politicians in the Pakistan parliament, and how to avoid opposition strikes. There are two documents that discussed amicable relationship between them. Some documents were about independent discussions of Imran Khan and Nisar with Taliban. Although we considered them non-relevant, we assert that there are elements of *discovery* within these documents. That is, Taliban as an emerging named entity in search for **about-ness** who is connected with two named entities in a query.

The summary lift in $F_1 - measure$ for English is shown below.

Sample Queries	$\boldsymbol{F_1 measure}$
q_1	0.07
$q_{1,ne}$	0.41

q_2	0.50
$q_{2,ne}$	0.77
q_3	0.41
$q_{3,ne}$	0.61
q_4	0.23
$q_{4,ne}$	0.70

Table 24

Results show that named entity recognition and matching of names is still a major issue in English despite major advances. The challenge in English is resolving the matched entities to the same entity and Identity matching (Lisbach 2015)(Round 2017). It is clearly demonstrated that identifying the named entities and indexing them without stemming improves IR system's performance. Evaluation shows that on average the F_1 measure score gets a lift of 28% with named entity aware search. Besides the improvement in IR system, discovery of new named entities can be triggered by a name-aware search.

6.5 Discussion

Analysis shows that recognition of named entities in an Information Retrieval system improves the results significantly especially for morphologically complex languages. It is important to note that we did the analysis on a commercial engine well known for its content and its global presence. It is clear that the named entity engines that are available for Russian for tagging are sub optimal due to the lack of and the quality of training data. In addition, there is no commercial impetus to invest in creating quality gazetteers or other enabling technologies for these morphologically complex languages because the cost will be higher than creating similar resources for non MRL. The quality of the named entity engines plays a significant role in improving search. We saw evidence of that as a number of the documents in English and Russian evaluation were either not classified correctly because of the machine learning based engine, or too broad of a taxonomy

scheme. That is, an in Russian evaluation there is not classification of cities. Therefore, the lack of coverage can become a problem.

Although it is important that the results of evaluations of each language should remain independent because of the corpus size, varying capabilities in enabling technologies etc., some of the commonly identified themes in the evaluation are:

- Transliteration is a major challenge for name identification as is supported in the literature.
- Stemming of named entities generally hurts the system performance. Over stemming in Arabic was a major challenge
- The usage of metadata to search for named entities should be dependent upon the accuracy the engine that put the tag.
- If the performance of the classification engine to identify named entities is suboptimal to be stored in metadata, query boosting can help in improving the rank. During the evaluation of Concept search in English, adding the weights to the query can help improve the **about-ness** of the search. Named Entity is an excellent candidate for boosting the query terms as is demonstrated that in the evaluation.
- Boosting of named entities can help in discovery.This phenomenon is observed in Urdu and English evaluation. In Urdu, a number of documents about *Azhar Masood* –a fundamentalist leader, and *Jaish-e-Muhammad* – a terrorist organization was retrieved in connection with *Omer Sheikh*. In English, we saw this during Pakistan's involvement in Yemen, and also two political parties negotiating with the Taliban.

The morphological rich languages pose a greater challenge than others for the creation of linguistic resources like stemmers, and IR systems because polysemy and synonymy and a robust NER system can help alleviate some of these challenges.

The figure below shows the average lift from baseline for Russian, Arabic, English and Urdu. The lift is shown for each metrics for the evaluation. The table clearly shows that there is a statistically significant in each of the evaluation categories. The Chi-squared

distribution shows the p-value of 0.0296 with the Chi-squared value of 38.675, rejecting the null hypothesis that the baseline is optimal Search.

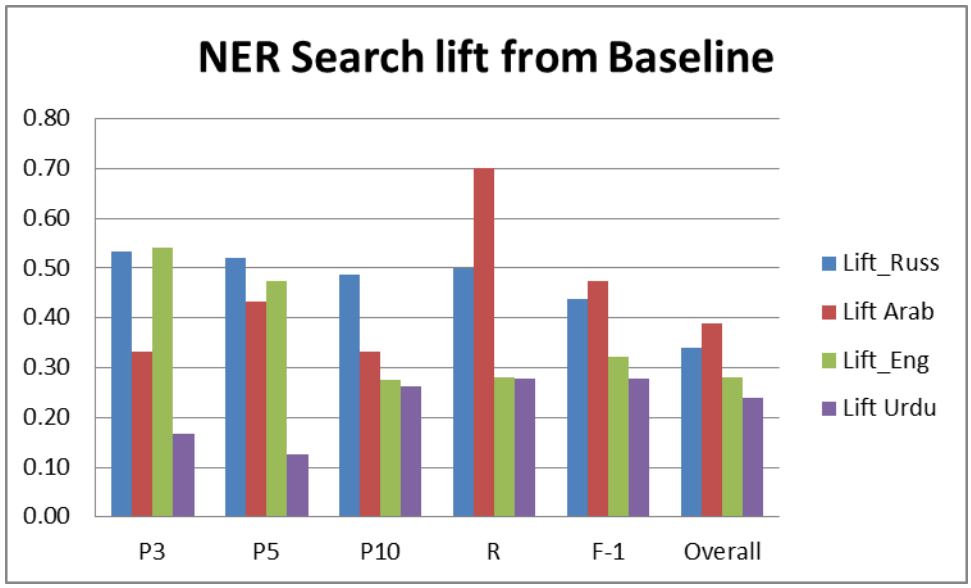


Figure 8

The table below shows improvement in each evaluation category for across all the languages evaluated.

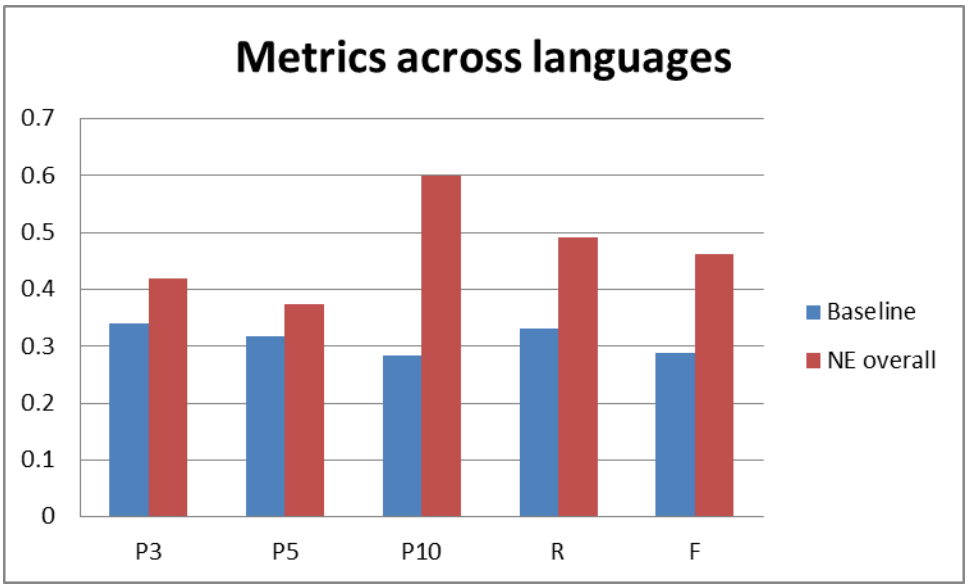


Figure 9

The summary of the performance of each language is shown below:

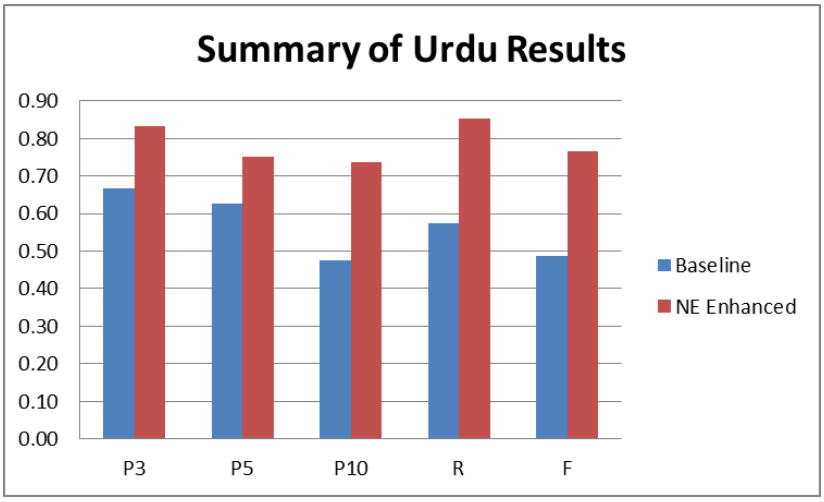


Figure 10

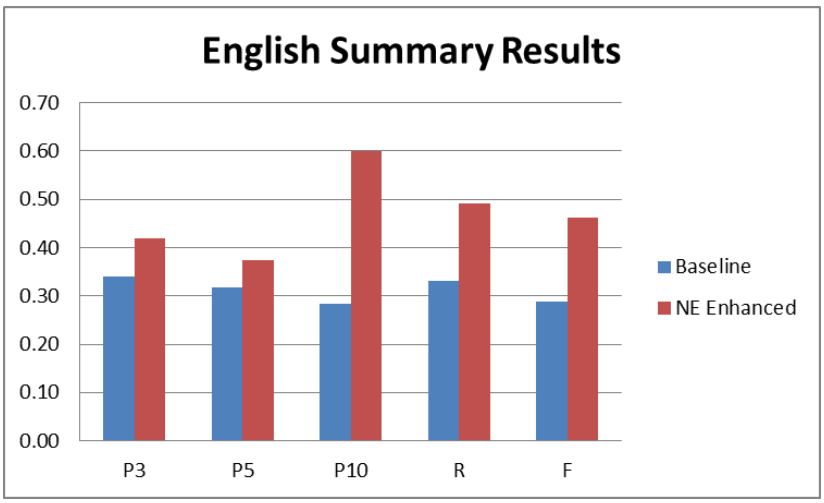


Figure 11

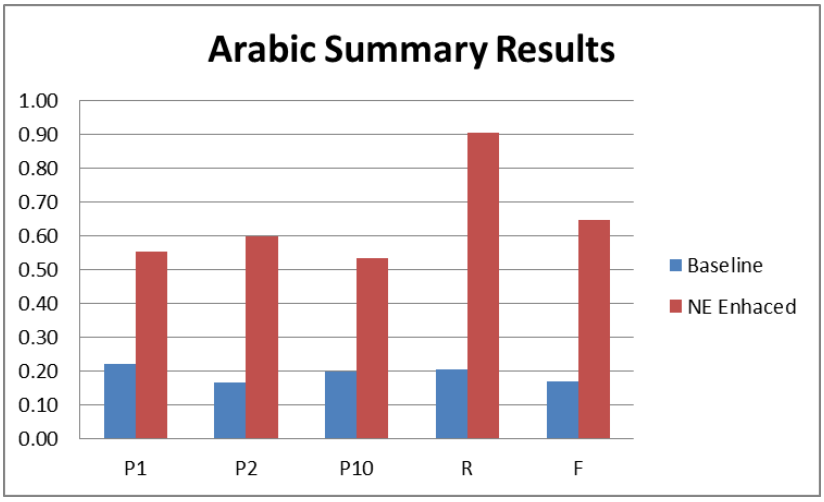


Figure 12

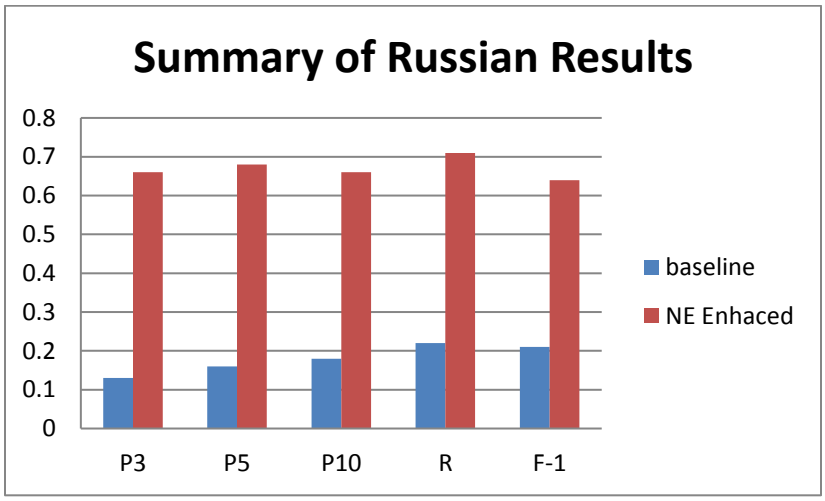


Figure 13

7 Conclusion & Future work

This research shows the utility of Named Entities in the field of Information Retrieval using Urdu as a case study for a morphologically rich and linguistic scarce language. Many contributions are made to the NLP community through the creation of Urdu linguistic resources and detailing the challenges associated with creating such resources for Arabic Script morphological rich languages. It is also shown the utility of NER for search improvements goes beyond MRL languages but also improves the Search for non-MRL

language like English which is considered the most used and most researched language in Search and NER. This research also shows that Name-aware stemmers and stop words will improve the search performance.

There is a great need for a quality named entity annotated corpus that contains a large set of documents, entities, and variations of word forms to research the viability of statistical and machine learning approaches.

This research has scratched the surface of the utility of NER on the MRL. In the future I intend to explore the discovery and relationships of related entities in a corpus. In addition, there is room to extend the weak stemmer to a full functioning robust stemmer.

8 References

1. Moldovan, R. Bot, G. Wanka. “Latenet Semantic Indexing for Patent Document.” International Journal of Apllied Mathematics and Computer Science 2005. Vol. 15. No. 4. pp 551-560. 2005.
2. Abney, S., Statistical Methods in Linguistics, In Klavans, J. & Resnik, P. (eds) The Balancing Act pp. 1-25, Cambridge, MA: MIT Press
3. Ahmed, Tafseer, and Annette Hautli. Developing a basic lexical resource for Urdu using Hindi WordNet. Proceedings of CLT10, Islamabad, Pakistan (2010).
4. Al-Halimi; Kazman; “Temporal indexing through lexical chaining”; Fellbaum, Christiane, ed., WordNet: An Electronic Lexical Database, MIT Press, May 1998.
5. Altszyler, Edgar & Sigman, Mariano & Fernández Slezak, Diego. (2016). Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database.
6. Bailey, G., "A History of Urdu literature--Urdu Poetry in Lucknow in the 19th century, 2018
7. Ballesteros, Lisa, and W. Bruce Croft, Phrasal translation and query expansion techniques for cross-language information retrieval." SIGIR Forum, Vol. 31. No. SI. ACM, 1997.
8. Bashir, E; Urdu and Linguistics: A Fraught but evolving relationship, Annual of Urdu Studies vol. 26, 2012
9. Berger and J. Lafferty; “Information Retrieval as Statistical Translation”; In Proceedings of SIGIR-99, Berkeley, CA, August 1999.
10. Berger, A., Della Pietra S.A, Della Pietra V. J., A Maximum Entropy Approach to Natural Language Processing, Computational Linguistics 22(1), 39-71, 1996
11. Bikel, D, M., Schwartz, R., Weischedel, R., What’s in a Name, Machine Learning, 34, 211-231,1999, Kluwer Academic Publishers.
12. Bresnan, J., Lexical-Functional Syntax. Blackwell, Oxford. 2001
13. Brill, E., A Simple Rule-Based part of Speech Tagger, Proceedings of the 3rd Conference on Applied Natural Language Processing, 112-116, 1992
14. Brill, E., Mooney, R.J., An Overview of Empirical Natural Language Processing, AI Magazine, v18 (4) 13-24, Winter (1997).
15. Fox. “Lexical Analysis and Stoplists”. Information Retrieval, Data Structure & Algorithms, pages 102-130, Prentice Hall, 1992.
16. H. Papadimitriou, P. Raghavan, H. Tamaski, S. Vempala, “Latent Semantic Indexing: A Probabilistic Analysis.” Proc. Of Symposium on Principles of Database Systems (PODS). Seattle, Washington. June 1998. ACM Press.
17. Cardie, C., Empirical Methods in Information Extraction, AI Magazine, v18 (4) 65-80, Winter (1997)
18. Cardie, C., Mooney, R.J., Machine Learning and Natural Language, Machine Learning, 34, 1-5, 1999, Kluwer Academic Publishers.
19. Charniak, E., Statistical Language Learning, MIT Press, 1993.

20. Charniak, E., Statistical Techniques for Natural Language parsing, AI Magazine, v18 (4) 33-44, Winter (1997)
21. Chen, A, Gey, F., “Building and Arabic Stemmer for Information Retrieval”, TREC2002.
22. Chieu, H., Ng, H., Named Entity Recognition: A Maximum Entropy Approach Using Global Information, COLING 2002.
23. Chomsky, N., Syntactic Structure, Mouton & Co. Publishers, Hague, 1957.
24. Church, Kenneth W, A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In Proceedings of ACL Conference on Applied Natural Language Processing, 136-143, 1988
25. 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.
26. 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.
27. D. Bikel, S. Miller, R. Schwartz, R. Weischedel, R. “Nymble: A High Performance Learning Name-Finder”, Proceedings of 5th Conference on Applied Natural Language Processing. 1996
28. D. MacMohan. “Linear Algebra Demystified.” McGraw Hill, 2006.
29. Dagan, I., Lillian L., Pereira, F., Similarity-Based methods for Word Sense Disambiguation, In Proceedings of ACL-EACL, 56-63, 97.
30. Daille, B., Study and Implementation of Combined Techniques for Automatic Extraction of Terminology, In Klavans, J. & Resnik, P. (eds) The Balancing Act, 49-66, Cambridge, MA: MIT Press
31. David D. Lewis: Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. European Conference on Machine Learning, 4-15, 1998
32. Davie, A., Maxell, M. Browne, E. Lynn, N., "Urdu Morphology", 2009. Technical Report. University of Maryland. https://www.casl.umd.edu/publications/urdu-grammar/ (Accessed, 2017)
33. Kareem Darwish, (2002). "Al-stem: A light Arabic stemmer" [Online]. Available: http://www.glue.umd.edu/~kareem/research.
34. Dozier, C., Assigning Belief Scores to Names in Queries, First International Conference on Human Language Resource, 2001
35. FIRE -- Forum for Information Retrieval Evaluation. http://www.isical.ac.in/~fire/talip.html (Feburary, 2010)
36. Fisher, D., Rilloff, E., Applying Statistical Methods to Small Corpora: Benefiting from a Limited Domain, In working notes of the AAAI Fall Symposium on probabilistic approaches to natural language 1992, pp- 47-53.
37. Ghuravi , Allama Aqeelul, YouTube “Topic : Mir Anees / Mirza Dabeer” Venue : Husainiya Ikramulla Khan, Nakhas Market, Lucknow, India January 2013

38. Goldsmith J. Higgins, D. Soglasnova, S. (2001) Automatic Language-Specific Stemming in Information Retrieval. Lecture Notes in Computer Science, Springer vol. 2069 pp273-284.
39. H. Bast. “An Existence Proof of the Singular Value Decomposition.” http://www.mpi-sb.mpg.de/~bast/seminar-ws04/lecture2.pdf. 2004.
40. H. Haav, T. Lubi. “A Survey of Concept-Based Information Retrieval Tools on the Web.” Fifth East-European Conference on Advances in Databases and Information Systems. 2002.
41. Hearst, M., Untangling Text Data Mining, In Proceedings of Association of Computational Linguistics, pages not available, June 1999, invited speaker.
42. http://al-bab.com/arabic-language/arabic-language
43. http://crl.nmsu.edu/Resources/lang_res/urdu.html
44. http://www.comp.lancs.ac.uk/computing/research/stemming/Links/evaluation.htm
45. http://www.lancs.ac.uk/fass/projects/corpus/emille/ (accessed 8/22/2013)
46. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.
47. Hussain, I; (Urdu ka Tahzibi Mizaj), [Urdu's Cultural Temperament]. The Annual of Urdu Studies 15(2):372–76. [Originally published in Akhbar-eUrdu (Islamabad) May 1998.]
48. Introduction to Modern Information Retrieval; Salton, G., McGill, McGraw-Hill, 1983
49. J. Savoy. “A Stemming Procedure and Stop word List for General French Corpora”. Journal of the American Society for Information Science, 1999.
50. Jay M. Ponte and W. Bruce Croft", "A Language Modeling Approach to Information Retrieval", SIGIR 99
51. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation
52. Jurafsky, D., Martin, J., Speech and Language Processing, Pearson Prentice Hall, 2nd edition, 2012
53. Kapur, S., Clark. R., The Automatic Construction of a Symbolic Parser via Statistical Techniques, In Klavans, J. & Resnik, P. (eds) The Balancing Act, 95-118, Cambridge, MA: MIT Press
54. Karine Megerdoomian, The Structure of Persian Names, MITRE Technical Report, 2008
55. Khormuji., M., Bazrafkan, M., “Persian Named Entity Recognition based with Local Filters, International Journal of Computer Applications (0975 8887) Volume 100 - No. 4, August 2014
56. Kontostathis, A; A Term Co-occurrence Based Framework for Understanding LSI: Theory and Practice. PhD Thesis. Department of Computer Science and Engineering, Lehigh University. 2003.
57. Lisbach, Meyer, "Linguistic Identity Matching", Springer, 2013

58. M. Berry, S. Dumais, G. O'Brien. "Using Linear Algebra for Intelligent Information Retrieval." Technical Report. University of Tennessee Knoxville. 1994.
59. M. Berry, S. Dumais, G. O'Brien. "Using Linear Algebra for Intelligent Information Retrieval." Technical Report. University of Tennessee Knoxville. 1994.M. Deniston. "An Overview and Discussion of Concept Search Models and Technologies." Engenium's Semetric (White Paper). 2003.
60. M. Deniston. "An Overview and Discussion of Concept Search Models and Technologies." Engenium's Semetric (White Paper). 2003
61. Manning, C.D., Raghavan, P., Schutze, H., Introduction to Information Retrieval, Cambridge University Press, 2012
62. Manning, C.D., Schutze, H. Foundations of Statistical Natural Language Processing, MIT Press, 1999
63. Matthews, D. Urdu in India, Annual of Urdu Studies vol. 17 (2002).
64. Miller. George, 1995. WordNet: a lexical database for English. Commun. ACM 38, 11 (November 1995), 39-41
65. Mitchell, T., Bluhm, A., Combining Labeled and Unlabeled Data with Co-training. Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers, 1998.
66. Mohamed Maamouri (project head), et al. Arabic Treebank: Part 3 (full corpus) v 2.0 (MPG + Syntactic Analysis) LDC2005T20. Web Download. Philadelphia: Linguistic Data Consortium, 2005
67. Mokhtaripour, A; Jahanpour, S. Introduction to a New Farsi Stemmer, CIKM 2006
68. Mooney, R., Inductive Logic Programming for Natural Language Processing, In Proceeding of the 6th International Inductive Logic Programming Workshop, pp. 205-224, Stockholm, Sweden, August 1996.
69. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. InAdvances in neural information processing systems 2013 (pp. 3111-3119).
70. Moulinier, I. "Thomson Legal and Regulatory at NTCIR-4: Monol-ingual and Pivot Language Retrieval Experiments". Working notes of NTCIR-4. Tokyo. 2004.
71. Moulinier, I; Jackson, P, Natural language Processing for Online Applications, Benjamins Publishing Company, 2002
72. N. Ide, C. Brew. "Requirements, Tools, and Architectures for Annotated Corpora". Proceedings of Data Architectures and Software Support for Large Corpora. European Language Resources Association, Paris, 2000.
73. Nagam, K., Ghani, R., Understanding the Behavior of Co-training., Proceedings of the Workshop on Text Mining at the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000).
74. Nayyer, R., Madni, F., Analysis of Intonation Patterns in Urdu, 2010, CLURP report.
75. Ng, H.T., Zelle, J., Corpus-based Approaches to Semantic Interpretation in Natural Language Processing, AI Magazine, v18 (4) 45-64, Winter (1997).
76. NIST http://www.nist.gov/tac/about/index.html 2017 (last updated)

77. NLP Overview http://language.worldofcomputing.net/nlp-overview/natural-language-processing-overview.html, 2017 (last updated)
78. Non-Nominative Subjects in Urdu: A Computational Analysis, with T.H. King in Proceedings of the International Symposium on Non-Nominative Subjects, ILCAA, Tokyo, December 2001, 525-548.
79. P. Baker, A. Hardie, T. McEnery, and B.D. Jayaram. “Corpus Data for South Asian Language Processing”. Procee-dings of the 10th Annual Workshop for South Asian Language Processing, EACL 2003.
80. P. Baker, A. Hardie, T. McEnery, and B.D. Jayaram. “Corpus Data for South Asian Language Processing”. Procee-dings of the 10th Annual Workshop for South Asian Language Processing, EACL 2003.
81. Palmer, D. Day, D. 1996, A Statistical Profile of the Named Entity Task, Proceedings of 5th Conference on Applied Natural Language Processing.
82. Porter M. (1980) An algorithm for suffix stripping. Program vol. 14: pp 130-137
83. Porter M. (2001) Snowball: A language for stemming algorithms. https://snowball.tartarus.org/texts/introduction.html
84. Press Trust of India, (2015, May 21st), My thinking process starts with my pen: Gulzar. http://indianexpress.com/article/entertainment/music/my-thinking-process-starts-with-my-pen-gulzar/
85. R. K. Belew. “Finding Out About”. Cambridge University Press, 2000.
86. R. Lo, B. He, I. Ounis. “Automatically Building a Stopword List for an Information Retrieval System”. 5th Dutch-Belgium Information Retrieval Workshop (DIIR). 2005.
87. R. Madala, T. Takenobu and T. Hozumi, The Use of WordNet in Information Retrieval. Montreal, Proc. Conf. Use of WordNet in Natural Language Processing Systems, pp. 31-37, 1998.
88. R. Story. “An Explanation of the Effectiveness of the Latent Semantic Indexing by Means of a Bayesian Regression Model.” Information Processing and Management. vol. 32, no. 3, pp 329-344. 1996.
89. Ratnaparkhi, A., Roukos, S., Ward, R.T., A Maximum Entropy Model for Parsing, In Proceedings of International Conference on Spoken Language Systems 803-806, 1994, Yokohama, Japan.
90. Ravin, Y.; Zunaid, Kazi; Is Hillary Rodham Clinton the President? Disambiguating names across documents;
91. Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing of morphologically rich languages (SPMRL): what, how and whither. In Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages (SPMRL '10). Association for Computational Linguistics, Stroudsburg, PA, USA, 1-12.
92. Riaz, K., Base-line for Urdu IR Evaluation., INEWS 08, Proceedings of the 2nd ACM workshop on Improving non English Web searching CIKM 2008.

93. Riaz, K., Challenges in Urdu Stemming" Future Directions in Information Access. Glasgow, August 2007
94. Riaz, K., Comparison of Hindi and Urdu in Computational Context, International Journal of Computational Linguistics and Natural Language Processing, volume 1, issue 3. 2012
95. Riaz, K., Concept Search in Urdu. PIKM08, Proceedings of the 2nd PhD workshop on Information and knowledge management, CIKM 2008
96. Riaz, K., Modern Approaches to Natural Language Processing, Conference on Emerging Technologies, Saint Paul, Minnesota, 2003
97. Riaz, K., Rule-based Named Entity Recognition in Urdu., Proceedings of the 48th Association of the Computational Linguistics Workshop on Named Entities. (2010)
98. Riaz, K., Stop Word Identification in Urdu", Conference of Language and Technology, Bara Gali, Pakistan, August 2007
99. Riaz, K., Urdu is not Hindi for Information Access., Workshop on Multilingual Information Access, SIGIR 2009.
100. Rich, E., Knight, K., Artificial Intelligence, McGraw-Hill, Inc. 1991.
101. Riezler, Stefan, et al., Statistical machine translation for query expansion in answer retrieval., Association of Computational Linguistics Vol. 45. No. 1. 2007.
102. Rizvi, J et. al. Modeling case marking system of Urdu-Hindi languages by using semantic information.", Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE '05.
103. Roth, D., Learning in Natural Language, IJCAI, 898-904,1999, best paper award.
104. Round, J, "Name Screening" A practical guide to computer based matching and manual elimination of name matches within a sanctions screening environment.
105. Ruff, Bart, Understanding Russian Names, Technical Report, Emporia State University. https://www.emporia.edu/~bartruff/docs/RussiaNames.pdf Accessed (2015)
106. Russell, S., Norvig, P., Artificial Intelligence, A Modern Approach, Prentice Hall Inc. Chapter 22 - p.15, 667, 1995
107. S. Deerwester, S. Dumais, T. Landauer, G. Furnas, R. Harshman. "Indexing by Latent Semantic Analysis." Journal of the American Society of Information Science," vol 41. no. 6. p. 391-407. 1990.
108. S. Deerwester, S. Dumais, T. Landauer, G. Furnas, R. Harshman. "Indexing by Latent Semantic Analysis." Journal of the American Society of Information Science," vol 41. no. 6. p. 391-407. 1990.
109. S. Leach. "Singular Value Decomposition—A Primer." Brown University. Unpublished Manuscript.
110. S. Palmer. D. Day, A Statistical Profile of the Named Entity Task, Proceedings of 5th Conference on Applied Natural Language Processing, 1996
111. Schmidt, R. "Urdu: An Essential Grammar." Routlege Publishing, 2005

112. Schmidhuber, J., Deep learning in neural networks: An overview *Neural Networks*, Volume 61, January 2015, Pages 85-117
113. Shaalan, Khaled. A Survey of Arabic named Entity Recognition and Classification, Computational Linguistics, Volume 40, no 2. 2014
114. Singal, A; "Modern Information Retrieval" IEEE Data Engineering, 2001
115. Syed AZ, Muhammad A, Martínez-Enríquez, "Sentiment Analysis of Urdu Language: Handling Phrase-Level Negation. In: Proceedings of the 10thMexican international conference of artificial intelligence, pp 382–393, 2011
116. Tsarfaty R, Seddah D, Kübler S, Nivre J. Parsing morphologically rich languages: Introduction to the special issue. Computational linguistics. 2013 Mar;39(1):15-22.
117. T. Landauer, P. Foltz, Lahman, D. "A Introduction to Latent Semantic Analysis." Discourse Processes, 25, 259-284. 1998.
118. Taghva K; Beckley, R; Sadeh M; A stemming algorithm for the Farsi language Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)
119. The Hidden Bias of Scene's Universal Language. The Atlantic, August 21, 2015
120. Visweswariah, Karthik, Vijil Chenthamarakshan, and Nandakishore Kambhatla. Urdu and Hindi: Translation and sharing of linguistic resource. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pp. 1283-1291. Association for Computational Linguistics, 2010.
121. W. Pottenger, A. Kontostathis. "A Framework for Understanding LSI Performance." Proceedings of ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval (ACMSIGIRMF/IR '03). 2003.
122. W0038: The EMILLE Lancaster Corpus. [cited 2013 January 1], Available: http://www.elda.org/catalogue/en/text/W0038.html
123. Wacholder, N., Ravin, Y., Choi, M., Disambiguation of Proper Names in Text, Proceedings of 5th Conference on Applied Natural Language Processing. 1996,
124. Workshop on Building and Using Parallel Texts for Languages with Scarce Resources at ACL 2005. http://www.cse.unt.edu/~rada/wpt05/ (February, 2010)
125. Workshop on NER for South Asian and South East Asian Languages at IJCNLP. http://ltrc.iiit.ac.in/ner-ssea-08/ (2010)
126. Yarowsky, D., Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In Proceedings of the 33rd Annual meeting of the Association of Computational Linguistics, 189-196, 1995.
127. Z. Xiao, A. McEnery, P. Baker and A. Hardie, "Developing Asian Language Corpora: Standards and Practice", Proceedings of the 4th Workshop on Asian Language Resources. March 25, 2004. Sanya, China.
128. Zaghouani, W. 2012. RENAR: A rule-based Arabic named entity recognition system. ACM Trans. Asian Lang. Inform. Process. 11, 1, Article 2 (March 2012), 13 pages.

129. Zizka, J., Bourek, A., Frey, L., TEA: A Text Analysis Tool for Intelligent Text Document Filtering, Third International Workshop on Text and Speech Dialogue, Brno, Czech Republic, September, 2000.

Appendix A

A. Technical Details of SVD

This section details the mathematical details of SVD. It is adequate to say that there exists a unique decomposition—proof can be obtained from Bast 2004.

A.1 Input Matrix

The input of SVD is a term-document matrix with dimension *t* x *d*. Let us call this matrix *A*. SVD breaks that system into three special matrices. Let the decomposition be represented the as product of three other matrices.

$$A = U \Sigma V^T$$

Equation 15

Where U and V have orthonormal columns (i.e. $UU^T = I$ and $V^T V = I$). Σ is a diagonal matrix of rank *r*. Therefore, the system has been reduced to rank *r*. The new matrices are of the following dimensions $U_{t\times r}$, $\Sigma_{r\times r}$, and $V^T_{r\times d}$, where *t* is the number of terms, and *d* is the number of documents, *r* is the rank of the system, *r* = rank(*A*) = min(*t*, *d*). The singular values are the diagonal entries of the matrix Σ are Σ = diag $(\sigma_{1,}\ \sigma_{2,}.....\sigma_n)$, $\sigma_i > 0$ for $1 \le i \le r$, and $\sigma_j = 0$ for $j \ge r+1$.

SVD can viewed as a system that is rotating the axes of the n-dimensional space, such that the first axis runs along the direction of largest variation among documents, the second dimension runs along the direction with the second largest variation, and this process continues until the rank is reduced to some value *r*, whereupon we stop this process. The matrices U and V represent the terms and documents in this new space respectively, and the diagonal matrix Σ contains the singular value of A in the descending order. Therefore, the i^{th} singular value on the diagonal represents the amount of variation along the i^{th} axis. Therefore, SVD can be seen as performing a linear combination of the vectors in the matrices.

We call the decomposition singular because U and V are matrices of left and right singular vectors, and Σ is the diagonal matrix of singular values. Values of Σ are constructed to be all positive and ordered in decreasing magnitude. If we keep only the k largest values in Σ, the rest of them are set to zero. By doing so, we only keep the first k columns of U and first k rows of V^T. Let these new matrices be U_k, Σ_k, and $V^T{}_k$.

The new approximate matrix $\hat{A}$ is represented as follows:

$$\hat{A}_k = U_k \, \Sigma_k \, V_k^T$$ **Equation 16**

The two diagrams show how the matrix is decomposed. Figure 9 shows how the SVD decomposed the matrix into three matrices when the rank is *r*. Figure 10 shows the effect when we choose first *k* dimensions. The darkened area represents the new reduced dimensions.

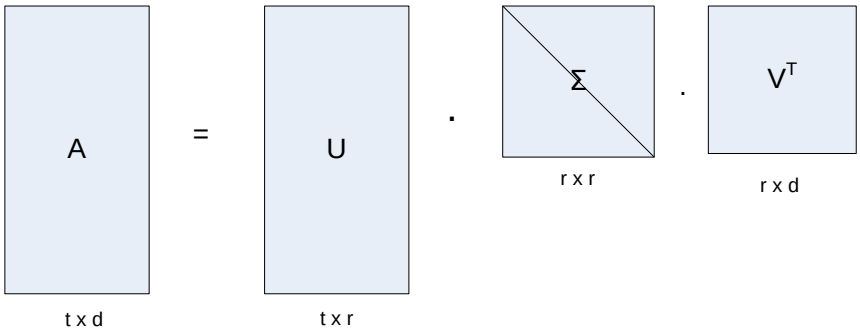


Figure 14

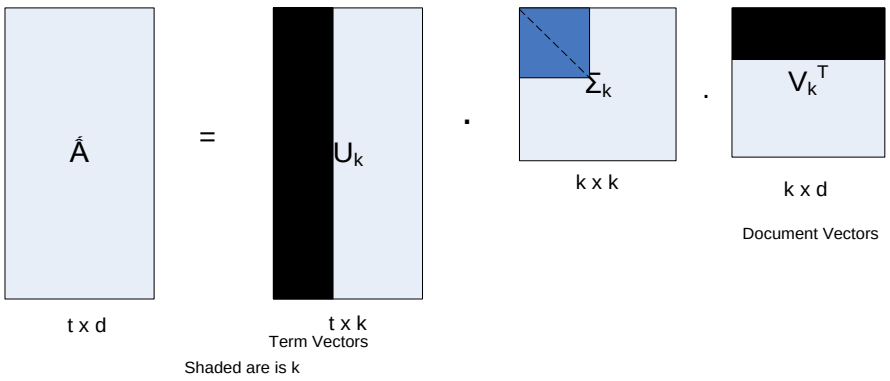


Figure 15

SVD is very closely related to eigen-analysis. In fact, U is the matrix of the eigenvectors of the square symmetric matrix of AA^T and V is the matrix of eigenvectors of $A^T A$, Σ^2 will be matrix of eigenvalues.

Singular value decomposition can be shown to satisfy the least-squares method property. When the new reduced matrix $\hat{A}_k$ is constructed, its representation in the original space is changed as little as possible when measured by the sum of the squares of differences. Specifically, SVD reduces the original matrix A to a lower dimension matrix $\hat{A}_k$ such that the 2-norm distance between the matrices is minimized. The 2-norm for matrices is the equivalent of Euclidean distance for vectors (Berry et al. 1994). Let that distance be denoted by:

$$\Delta = || A - A_k ||_2$$ **Equation 17**

From the above discussion, we can see that the goal of pure SVD is to reduce noise in a large data set and also to reduce the linear dependency of the system.

A.2 Queries

The user's query on the k-reduced dimension should be represented in the k-dimensional space as a vector and compared to the documents. As previously mentioned, a query is treated as a document and is referred to as a pseudo-document. A user's query can be mapped into the k-dimensional space as:

$$\hat{q} = q^T U_k \Sigma_k^{-1}$$

Equation 18

where q is the query vector that contains the user issued terms as its component (some weight could be associated with each component. Weighted sum is calculated through $q^T U_k$ and the right multiplication to Σ_k^{-1} differentially weights the separate dimension.

Therefore, the query vector when projected is located at the weighted sum of the constituent term vectors, or in other words, at the centroid of its corresponding term dimensions.

This query vector is then compared in similarity to each document vector in the system and the documents above a certain threshold are returned.

A.3 Updating SVD

Assume that we have computed SVD, and we have three unique matrices of the reduced dimension. If we receive new documents, how do we add these documents into our decomposed system and still maintain the stability of the system? One solution is to recompute SVD. This is an impractical solution because of computation costs. Berry (1994) proposed a method of in place folding of the new documents into the reduced system. This method is called *folding-in.* Through this process, a new document is folded in without effecting the pre-existing representation of the reduced system. Therefore, it can have a deteriorating effect on the accuracy of the reduced dimensions.

Example of SVD Computation:

Here we present a small example which we used to illustrate vector space model in Chapter 3. The input data is presented again for convenience. Given the matrix below, will be decomposed in the following manner.

	Apple	Animal	Zebra
Doc 1	0	3	7
Doc 2	10	1	2
Query	10	0	0

Table 25

$$A = \begin{pmatrix} 0 & 10 \\ 3 & 1 \\ 7 & 2 \end{pmatrix} \quad U = \begin{pmatrix} 0.91 & 0.43 \\ 0.18 & -0.35 \\ 0.39 & -0.83 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 10.51 & 0 \\ 0 & 7.25 \end{pmatrix} \quad V^T = \begin{pmatrix} 0.31 & 0.95 \\ -0.95 & 0.31 \end{pmatrix}$$

If we reduce the above system to k=1. We will get the following information.

$$U_{k=1} = \begin{pmatrix} 0.91 \\ 0.18 \\ 0.39 \end{pmatrix} \qquad \Sigma_{k=1} = \begin{pmatrix} 10.51 \end{pmatrix} \qquad V_{k=1}^T = \begin{pmatrix} 0.31 & 0.95 \end{pmatrix}$$

With these new reduced values we get the following new matrix:

$$\hat{A}_{k=1} = \begin{pmatrix} 2.9306 & 9.0511 \\ 0.5777 & 1.7843 \\ 1.2503 & 3.8616 \end{pmatrix}$$

Now we project the query into our reduced space by using equation 18, and we get the following:

$$\hat{q} = (0.869)$$

When we calculate the similarity of the query for each document vector, the score returned for Doc-1 is 0.2652 and score for Doc-2 is 0.891. Therefore, we pick Doc-2 as the document that best matches the query. The example shown here is crafted to be simple to show to the process, not to prove any results.

A.3 Other Comparisons

Deerwester (1990) showed that estimated matrix $\hat{A}$ gives us the opportunity to compute term-term and document-document comparisons. The product between two rows of $\hat{A}$ reflects the extent to which two terms have a similar pattern of occurrence across a set of documents. The matrix $\hat{A}\hat{A}^T$ is a square matrix containing all the term-to-term dot products. Since Σ is diagonal and V is orthonormal, we can verify derivation below:

$$\begin{aligned}\hat{A}\hat{A}^T &= U\Sigma V^T(U\Sigma V^T) \\ &= U\Sigma V^T(V^T)^T\Sigma^T U^T \\ &= U\Sigma V^T V\Sigma U^T \\ &= U\Sigma I\Sigma U^T \\ &= U\Sigma^2 U^T\end{aligned}$$

Therefore the (*i*,*j*) cell of $\hat{A}\hat{A}^T$ can be obtained from taking the dot product between the *i*th and *j*th rows of the $U\Sigma$ matrix. This can be done by considering the rows of $U\Sigma$ as coordinates of terms, and then the dot product between two such points (vectors) in space gives the comparison between terms represented in those vectors. Note that we use $U\Sigma$ as the coordinates instead of U. Since Σ is diagonal, the positions of the point are same except each of the axes has been shrunk or stretched according to the values of the diagonal elements of Σ. We can also view this through eigenvector representation. Since the columns of U represent the eigenvector and the values in Σ represent the eigenvalues of the corresponding vectors, the values of Σ are simply the scaling of the eigenvectors of U.

We can also show document-document similarity can be computed alike to term-term similarity, but instead of comparing two rows of $\hat{A}$, we are taking the dot product of two column vectors of $\hat{A}$. Such a comparison tells us the extent to which two documents have a similar profile of terms. Given our definition of matrices we can show that $\hat{A}^T\hat{A} = V\,\Sigma^2 V^T$. The argument for using $V\Sigma$ for computing the values of $cell\ (i, j)$ is same as above.

Appendix B

B.1 Alternate Views of LSA

B.1.1 Cognitive View

Landauer et al. (1998) argue that LSA extracts and represents the contextual meaning of the word by aggregating all the word contexts in which a word appears and does not appear. These set of mutual constraints determine the similarity of the meaning of words to each other. They view LSA to be exhibiting the following phenomena. In their words, "Given some text (corpus), there is so much of concept *1* and so much of concept *2* represented in the corpus. Generally, given a word *W,* it has so much of concept *1* and so much of concept *2*. Now by manipulating these two data by some vector algebra, my best guess is that word *W* actually appeared 0.6 (say) times in document Y." The authors argue that LSA induces knowledge similar to what a human being would learn. They claim that this learning and representation of words in the reduced dimension is comparable to a well-placed eighth grader. They show this by performing experiments with automatic essay grading and solving multiple choice questions.

B.1.2 Bayesian Regression View

Story (1996) very interestingly compares the latent semantic model to the Bayesian regression model. It is shown that the vector space model is equivalent to the statistical multiple regression model and then shows that the latent semantic model is closely related to, but not quite identical to the extension of multiple regression models that incorporates a limited application of Bayesian methods. Hence, LSA models work better than VSM in two ways. First, they remove certain information that is not reliable in a well-defined statistical model (noise removal), and secondly, LSA models reduce the magnitude of what amounts to specification error from the perspective of the related regression model. Both the criteria are satisfied in essence by replacing the insignificant singular values from the diagonal singular value matrix to zeros. Note that the second criterion is akin to relevance feedback in the information retrieval literature and can be considered as a something known a priori.

B.1.3 Term Co-Occurrence Analysis

Kontostathis et al. (2003) try to theoretically analyze LSI performance by focusing on term co-occurrence in the corpus. They analyze the term-term similarity matrix (i.e. the $U\Sigma$ matrix by categorizing the term-occurrence in a hierarchical structure). For example, if two terms co-occur in a document, they are found to have a direct relationship with each other, and they are categorized as first order term co-occurrence. If the terms that are not found in the same document but share a common term that is first order of co-occurrence to them, then they are categorized as a second order term co-occurrence and so on. We can illustrate the point with an example. Let D1, D2, and D3 be documents, and let *cat, dog, animal*, and *wolf* be the terms in some corpus. We show the arrangement of the term and documents below:

D1: cat, dog

D2: dog, animal

D3: animal, wolf

The order of co-occurrence is defined as the following:

1st order of co-occurrence: {cat, dog}, {dog, animal}, {animal, wolf}

2nd order of co-occurrence: {cat, animal}, {dog, wolf}

3rd order of co-occurrence: {cat, wolf}

The structure can be shown as the hierarchical graph where the order of co-occurrence is *n*+1 where *n* is the number of hops needed to connect the nodes in the graph.

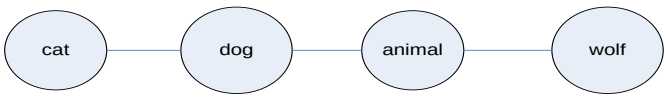


Figure 16

They show that the performance of SVD is directly dependent upon its ability to find the higher order co-occurrence terms. This was shown by analyzing the correlation coefficients for a number of collections. The process they used is somewhat related to a well-established data mining technique of association analysis.

Appendix C

Two queries from a commercial Search engine that support Boolean queries. The intent is find information **about** labor laws (a proper name) and trying to find legal names in the US cases with particular labor-related terms in the title. The theory here is that cases with these terms are likely to be labor-related just based on the parties involved.

(TI(A.F.L-C.I.O A.F.S.C.M.E BROTHERHOOD "CIVIL SERVICE" (DEPARTMENT DEPT +2 INDUSTRY) (DISTRICT TRADE +2 COUNCIL) E.E.O.C EMPLOYE EMPLOYER EMPLOYEE EMPLOYMENT (EQUAL +3 OPPORTUNITY) "HUMAN RESOURCES" I.A.M (INDUSTRIAL +2 CLAIM INSURANCE RELATIONS SAFETY WELFARE) (INTERNATIONAL +3 ASSOCIATION) LABOR LABORER) % TI(IAMS "LABOR PARTY"))

(TI(LOCAL LONGSHOREMAN "MERIT SYSTEM" N.L.R.B PERSONNEL (UNION /1 INTERNATIONAL PILOT TRANSPORTATION WAREHOUSEMAN) STEEL TEAMSTER "TRAFFIC CONTROLLER" U.A.W (UNION +1 NO NUMBER) "VOCATIONAL REHABILITATION" (WAGE /3 AGREEMENT APPEAL BOARD CLAIM! DECEASED DETERMINATION EARNER HOUR MINIMUM PREVAILING STABILI!) "WORK! COMPENSATION" WORKERS) % TI("LOCAL TELECOMMUNICATION"))