



UNIVERSITY OF NAIROBI

College of Biological and Physical Sciences

School of Computing and Informatics

AUTOMATIC CHARACTERIZATION OF NAMED ENTITY RELATIONAL FACTS IN UNSTRUCTURED INCIDENT REPORTS

By

EDWIN KIMATHI IKUNYUA

REG. NO. P58/61712/2010

SUPERVISOR: LAWRENCE MUCHEMI

August 2012

Project report submitted in partial fulfillment of the requirements for the
award of a degree in Msc. Computer Science

ACKNOWLEDGEMENT

I wish to acknowledge my supervisor Mr. Lawrence Muchemi, panel members Dr. Christopher Moturi, Prof. Peter Wagacha, Mr. Daniel Orwa and Mr. Joseph Ogutu for their expert supervision and advice through out my project work.

My sincere gratitude goes to: my family, friends and colleagues, for their faith, enthusiasm and friendship; my parents for being wonderful teachers and examples.

Finally, I wish to thank God for: health, strength and sanity to do what had to be done.

DECLARATION

The project presented in this report is my original work and it has not been presented to any institution of higher learning for the purpose of academic evaluation.

Name: Edwin Kimathi Ikunyua

Signature: ..... **Date:** 12/11/12

This project has been submitted as a partial fulfillment of the requirements of the Master of Science degree in Computer Science at the University of Nairobi with my approval as the University Supervisor.

Name: Lawrence Muchemi

Signature: ..... **Date:** 12/11/2012

DEDICATION

To all Kenyans who endeavor for success by offering true value to society and rightfully understand that reputation is built on actions not mere intentions. Is that you?

ABSTRACT

Natural language provides many different ways of expressing facts. These facts can either be explicit facts or implicit facts. Explicit facts could be in the form of entity relations expressed in a single sentence. Many organizations own document corpuses that take the form of unstructured Incident Reports, which contain explicit facts. A key challenge faced by these organizations is finding out how two named entities contained in a unstructured Incident Report corpus are related to each other; a reading problem.

In this research we conceptualized the problem as a composition of two sub problems; relational extraction and relational representation. We used Open Information Extraction tools and techniques to extract Entity Relational facts; a dictionary of named entities and a greedy algorithm to tag and characterize the extracted facts and graph algorithms to search through the extracted facts to determine the interrelationship between two (2) named entities in a Test corpus of ten (10) documents covering Politics, Accidents and Poaching.

We came up with a model that harmonizes relation extraction and representation, which was able to address the key challenge of being able to determine how two named entities are interrelated in a unstructured Incident Report corpus.

From experiments conducted using a prototype application developed based on the model above it was observed that: the quality of the text corpus, the choice of the underlying POS tagger and English dictionary, the character and size of Named Entity Dictionary and a mechanism to enable document level named entity resolution are key issues that have to be addressed when building a Entity Relation Characterizer.

The model developed is a useful tool that can guide in the development of systems that collate information containing named entity relational facts from different sources, addressing the issue of information incoherence within organizations.

Keywords: Natural Language, Relation Extraction, Graph, Named Entity, Information Extraction, Corpus.

TABLE OF CONTENTS

ACKNOWLEDGEMENT	II
DECLARATION	III
DEDICATION	IV
ABSTRACT	V
TABLE OF CONTENTS	VI
LIST OF TABLES.....	VIII
LIST OF FIGURES.....	IX
ABBREVIATIONS.....	IX
DEFINITION OF TERMS	IX
CHAPTER ONE: INTRODUCTION.....	1
1.1 Background.....	1
1.2 Problem definition	2
1.3 Objectives	3
1.4 Research Question	3
1.5 Research outcomes	3
1.6 Assumptions and limitations of the Research.....	4
CHAPTER TWO: LITERATURE REVIEW	5
2.1 An overview of natural Language	5
2.2 Information Extraction (IE).....	6
2.3 Entity Relational Facts.....	6
2.4 Approaches to Relation Extraction.....	8
2.4.1 Supervised approach to Relation Extraction	8
2.4.2 Semi Supervised Approaches to Relation Extraction.....	9
2.4.3 Unsupervised Approaches to Relation Extraction.....	10
2.5 Comparative Analysis of alternative approaches to Relation Extraction	14
2.6 Relational Representation Using Graphs.....	15
2.7 Related work.....	18
2.8 Conceptual Framework.....	20
CHAPTER THREE: METHODOLOGY	22
3.1 Research Design	22
3.1.1 Sampling Procedure.....	23

3.1.2 Observation Procedure	23
3.1.3 Statistical Procedure	24
3.1.4 Experimental Procedure	25
3.1.5 Justification for the Design.....	26
3.1.6 Sources of data and relevance of data to the problem	26
3.1.7 Procedures and methods for data collection.	27
3.1.8 Data analysis methods and their justification	28
3.2 System Design and Implementation	29
3.2.1 System Architecture	30
3.2.2 Component Description.....	31
3.2.3. System Test Plan	39
3.2.4 System Implementation	40
3.3 Limitations of Methodology	41
CHAPTER FOUR: RESULTS AND ANALYSIS.....	42
4.1 Creation of a model Test corpus.....	42
4.2 Design of Entity Relation Characterizer.....	43
4.3 Prototype Development	43
4.4 Prototype Evaluation	43
4.4.1 Experiment 1.....	44
4.4.2 Experiment 2.....	46
4.4.3 Experiment 3.....	51
4.4.4 Experiment 4.....	56
4.4.5 Experiment 5.....	70
CHAPTER FIVE: DISCUSSION	74
5.1. Conclusion.....	74
5.2 Limitations.....	75
5.3 Recommendations for further Research	75
REFERENCES	76
APPENDIX 1: Part of Speech Tags	78
APPENDIX 2: News Stories in Test Corpus	79
APPENDIX 3: Program Code	86
APPENDIX 4: User Manual.....	93

LIST OF TABLES

TABLE 1: ENTITY RELATIONS TAGGING GUIDELINES (SOURCE ACE 2005)	7
TABLE 2: TAXONOMY OF BINARY RELATIONSHIPS	11
TABLE 3: SYNTACTIC CONSTRAINT RELATION PHRASE POS TAG PATTERN.	12
TABLE 4: ReVERB OUTPUT	13
TABLE 5: COMPARATIVE ANALYSIS OF ALTERNATIVE APPROACHES TO RELATION EXTRACTION	14
TABLE 6: EMAIL AND MEETINGS NODE AND RELATION TYPES	16
TABLE 7: SUMMARY OF ANNOTATIONS BY A & B	24
TABLE 8: SUMMARY OF HUMAN ANNOTATED AND PROTOTYPE ANNOTATED EXTRACTIONS	27
TABLE 9: SAMPLE OUTPUT OF RELATION EXTRACTOR	32
TABLE 10: SAMPLE SENTENCE WITH CORRESPONDING POS TAGS AND CHUNK TAGS	33
TABLE 11: SYSTEM DATABASE STRUCTURE	36
TABLE 12: SYSTEM TEST PLAN	39
TABLE 13: DESCRIPTION OF NEWS STORIES INCLUDED IN TEST CORPUS	42
TABLE 14: EXPERIMENT 1 OUTPUT- NAMED ENTITY SUGGESTIONS	45
TABLE 15: EXPERIMENT 2 PARAMETERS- NAMED ENTITIES AND THEIR ENTITY TYPE	46
TABLE 16: EXPERIMENT 2 OUTPUT - INVERTED INDEX (DETECTED ENTITIES)	48
TABLE 17: EXPERIMENT 2 OUTPUT - EXTRACTED ENTITY RELATIONS	49
TABLE 18: EXPERIMENT 2 OUTPUT- NAMED ENTITY SUGGESTIONS	50
TABLE 19: EXPERIMENT 3 OUTPUT - INVERTED INDEX (DETECTED ENTITIES)	52
TABLE 20: EXPERIMENT 3 OUTPUT - EXTRACTED ENTITY RELATIONS	53
TABLE 21: EXPERIMENT 3 OUTPUT- NAMED ENTITY SUGGESTIONS	54
TABLE 22: EXPERIMENT 3- INCORRECT ENTITY RELATION CHARACTERIZATION	55
TABLE 23: NATION NEWSPAPER NEWS STORY OF 6 TH JUNE 2012	57
TABLE 24: ANNOTATOR A & B ANNOTATION OF NATION NEWSPAPER STORY OF 6 TH JUNE 2012	58
TABLE 25: ANNOTATOR AND PROTOTYPE EXTRACTIONS OF THE NATION NEWS STORY OF 6 TH JUNE 2012	59
TABLE 26: NATION NEWSPAPER NEWS STORY OF 17 TH MARCH 2012	60
TABLE 27: ANNOTATOR A & B ANNOTATION OF NATION NEWSPAPER STORY OF 17 TH MARCH 2012	60
TABLE 28: ANNOTATOR & PROTOTYPE EXTRACTIONS OF THE NATION NEWS STORY OF 17 TH MAR 2012	61
TABLE 29: NATION NEWSPAPER NEWS STORY OF 18 TH APRIL 2012	62
TABLE 30: ANNOTATOR A & B ANNOTATION OF NATION NEWSPAPER STORY OF 18 TH APRIL 2012	63
TABLE 31: ANNOTATOR & PROTOTYPE EXTRACTIONS OF THE NATION NEWS STORY OF 18 TH APRIL 2012	64
TABLE 32: AN EXCERPT OF THE STAR NEWSPAPER STORY OF 23 RD JUNE 2012	65
TABLE 33: ANNOTATOR A&B ANNOTATION OF THE THE STAR NEWS STORY EXCERPT OF 23 RD JUNE 2012	65
TABLE 34: ANNOTATOR & PROTOTYPE EXTRACTIONS OF THE STAR NEWS STORY OF 23 RD JUNE 2012	66
TABLE 35: EXPERIMENT 4 OUTPUT - INVERTED INDEX (DETECTED ENTITIES)	67
TABLE 36: EXPERIMENT 4 OUTPUT - EXTRACTED ENTITY RELATIONS	67
TABLE 37: SUMMARY OF EXPERIMENT 4 RESULTS	68

TABLE 38: LIST OF OMITTED EXTRACTIONS	68
TABLE 39: LIST OF ERRORNEOUS EXTRACTIONS	69
TABLE 40: EXPERIMENT 5 OUTPUT - ENTITY LINK SEARCH	71

LIST OF FIGURES

FIGURE 1: AN EXAMPLE SUB-GRAPH, SHOWING THE CONNECTING PATHS	17
FIGURE 2: DIAGRAMMATIC REPRESENTATION OF THE CONCEPTUAL FRAMEWORK	21
FIGURE 3: EVOLUTIONARY PROTOTYPING PROCESS	29
FIGURE 4: OVERALL SYSTEM ARCHITECTURE	30
FIGURE 5: REVERB COMPONENTS.....	31
FIGURE 6: STRUCTURE OF NAMED ENTITY DETECTOR	32
FIGURE 7: SYSTEM DATABASE STRUCTURE	37
FIGURE 8: PROTOTYPE SYSTEM MAIN MENU.....	43
FIGURE 9: PROTOTYPE SYSTEM FILE READER.....	44
FIGURE 10: PROTOTYPE SYSTEM -ADD NEW NAMED ENTITY WINDOW	47
FIGURE 11: PROTOTYPE SYSTEM- ENTITY SEARCH DIALOG	70
FIGURE 12: GRAPH REPRESENTATION OF EXTRACTED ENTITY RELATIONS CONTAINED IN TABLE 36	72
FIGURE 13: IDEAL GRAPH REPRESENTATION TABLE 36 ENTITY RELATIONAL FACTS.....	73

ABBREVIATIONS

NL	Natural Language
NLP	Natural Language Processing
IE	Information Extraction
IR	Information Retrieval
IAA	Inter-annotator Agreement
NLTK	Natural Language Toolkit
POS	Part Of Speech
NED	Named Entity Detector
SSADM	Structured Systems Analysis And Design
PIM	Personal Information Management

DEFINITION OF TERMS

Unstructured data: - data whose properties and relationships are not obvious

Natural Language (NL):- a human written or spoken language used by a community

CHAPTER ONE: INTRODUCTION

1.1 Background

Natural Language (NL) is a central pillar in human civilization it serves as a basis of enabling cooperation, coordination and sharing of ideas between people. NL exists in two main forms; speech and written text; which complement each other, with speech enabling verbal communication and written text serving as a fundamental repository of human knowledge and understanding(Etzioni et al., 2011).

Many modern organizations own unstructured text corpuses that are reflective of the industry or mandate that they serve. With technological advancement the amount of text readily accessible to these organizations has long surpassed the ability of human beings therein to read it. In this study we focus on Incident Reports written in English, which emanate from various sectors such as: Insurance, Security, Media and Humanitarian Relief Agencies. Unstructured Incident Reports may exist in various forms in the afore mentioned sectors such as; Investigation reports in the insurance industry, Occurrence Book entries in the Security Industry, and news stories in the Media Industry. We define Incident Reports as professionally authored text documents written in Natural language which contain answers to wh-questions (Who, Where, When, Why, What), in form of Named Entities. Named Entities are definite noun phrases that refer to specific types such as Person, Location, Date, Facility, Geopolitical Entities (Countries, Counties etc), Organizations or Objects (e.g. Weapons etc).

As organizations seek to master their Incident Reports corpuses and address the gap of quantity of information vis-à-vis ability to read, they have relied on Keyword Search Mechanism and Supervised Information Extraction techniques with modest levels of success. However a key need of how two named entities are linked to each other within an unstructured text corpus still remains unaddressed.

To address this need organizations have sought to refine keyword search mechanism to include Boolean expressions. The amount of time and effort required to determine how two named entities are interlinked still depends on the sparseness of the relation between two entities in the corpus and the frequency of occurrence of the named entities in the corpus; limiting a user's

productivity to their ability to structure effective queries and the effectiveness of the underlying search mechanism.

In this study we use Open Information Extraction according to Banko et al. (2007) single learning model of how relations are expressed in English and Machine Reading - the automatic unsupervised understanding of text (Etzioni et al., 2006) to enable the extraction of entity relations from unstructured text. Further, we use graph theory to model pair wise relations between entities using the extracted relations, with graph nodes representing named entities and vertices representing relations between them. Finally we use graph algorithms such as Floyd's algorithm to find the transitive closure of the graph and identify how two named entities are interlinked.

1.2 Problem definition

In Natural Language there are many different ways of expressing facts in unstructured text. However mechanisms to access these facts are limited, with vast amounts of information existing in unstructured text collections primarily accessible through keyword querying at the document level, which ignores valuable relations found in underlying lexical and semantic relationships between terms and entities in the text (Agichtein and Cucerzan, 2005).

Individual unstructured Incident Reports in the Insurance, Security, Media and Humanitarian Relief Agencies carry a handful of facts that may help one understand a larger phenomenon, if read with other associated reports containing interlinked or related facts. These facts may be in various forms, among them Entity relations. A key challenge that faces users with a large unstructured Incident Report corpus presented in the conventional page view format and relying on keyword querying is *finding out how two named entities are related to each other in that text corpus*. Two alternatives exist:

1. Search for documents that contain the two named entities, read through the result to find out if there is a relation(s) that exist between the entities.
2. Search for documents that contain the named entities individually combine the documents and read through them to establish the existence of a relation.

The amount of time and effort required in both cases depends on the sparseness of the relation between the two entities in the corpus and the frequency of occurrence of the named entities in

the text corpus. The bounds of a user's productivity in this task is determined by their reading speed, their ability to structure effective queries and the effectiveness of the underlying search mechanism.

This raises the question, is it possible to employ machine reading techniques to automatically characterize Incident Reports in a manner that linearly scales the time required to explore relations between two named entities?

1.3 Objectives

The objectives of this study are:

1. To create a Test corpus for use by the Prototype application.
2. To design an entity relation characterizer.
3. To develop a prototype of the Entity Relation Characterizer.
4. To evaluate the performance of the prototype on the model text corpus.

1.4 Research Question

What Key issues should be addressed in order to develop an Entity Relation characterizer?

1.5 Research outcomes

Currently the success of investigating through an Incident Report corpus largely depends on an individual investigator's ability to read a large number of documents and identify entity relations between them; this success is largely anchored on an individual's ability to recall, organize and join simple facts. Further an investigator's efficiency and productivity is determined by their level of alertness and memory of seemingly trivial facts. This makes it difficult for two independent investigators looking at the same set of documents to arrive at the same conclusion. The adoption of this solution is expected to standardize the process of investigating through an Incident Report corpus.

At the organizational level the adoption of the outcome of this research is expected to aid organizations overcome the problem of information incoherence that arises from having different facts spread over many documents, whereby no single document has all the answers.

On a broader scale, this study will document the process of developing an entity relational characterizer that facilitates machine reading, with a view of enhancing the body of knowledge on unstructured text characterization.

1.6 Assumptions and limitations of the Research

The research assumes that organizations aiming to use the system developed have a fairly large corpus of professionally authored unstructured Incident Reports that contain wh-patterns. We further assume that the organizations have structured lists of named entities that they would wish detected and extracted from their text corpus.

The key limitation of this research is that it focuses on the extraction and representation of facts expressed in a single natural language sentence. This implies that facts that span multiple sentences may not be extracted. Additionally some facts expressed in a single sentence may also not be extracted since the Open Information Extraction paradigm adopted for the relation extraction component of this research covers approximately 95% of English binary relation. See Table 2.

CHAPTER TWO: LITERATURE REVIEW

For us to get a good understanding of the problem statement we divided the problem into two sub problems; Relational Extraction and Relation Representation. This literature review is organized as follows; the first section provides an understanding of natural language, what Entity Relational facts are, and the various approaches used to extract relations from natural language texts and a discussion of their strengths and weaknesses, the section concludes with a comparative analysis of the various approaches to Relation Extraction. The second section reviews the use of graphs data structures in representing relational information. We conclude the literature review by deriving a conceptual framework to solve the two sub problems.

2.1 An overview of natural Language

Akmajian et al. (2001) define Language as a conventional system for communication, a system for conveying messages. Syal and Jindal (2007) argue that language can be characterized as a system of systems, whereby sounds are arranged in a certain fixed or established systematic order to form meaningful units or words. Words are in turn arranged in a particular system to frame acceptable and meaningful sentences. The systems operate at two levels phonological and syntactical. At the phonological level sounds of a language appear in some fixed combinations. At the syntactic level words combine to form sentences according to certain conventions (grammatical and structural rules) of the language. Further, Syal and Jindal (2007) point out that the system can viewed as an hierarchy where units are made up of smaller units (the smallest unit being a phoneme) with rules that permit the occurrence and combination of smaller units. Communication is accomplished in the system only because words have certain meanings; meanings can be of two types: speaker meaning and linguistic meaning. Speaker meaning could be literal or non literal (meaning something different from what words mean e.g. sarcasm or irony) Akmajian et al. (2001).

The aforementioned system and the knowledge of language are used in designing and developing Natural language Processing Systems. Alternative arguments on what constitutes the knowledge of language exist, see Mills (2007), we however adopt the view of Jurafsky and Martin (2003) of knowledge of language, which states that the English language is composed of eight parts-of-speech (POS): noun, verb, pronoun, proposition, adverb, conjunction, adjective and interjection. The significance of which, is that POS give a significant amount of information about the word and its neighbors.

In this research we focus on Incident Reports which originate from trained writers and contain various facts relating to: Who, What, When, Where and Why. We adopt the view of Bagga (2000) who argues that, any text document is a collection of facts, which may be explicitly or implicitly stated and are therefore “easy” or “difficult” to comprehend. Easy facts may be found in a single sentence such as a city name. Difficult facts on the other hand facts are spread across several sentences (example: the reason for a particular event).

2.2 Information Extraction (IE)

Cowie and Lehnert (1996) define IE as a process, which takes unseen texts as input and produces fixed-format, unambiguous data as output. Robert and Wilks (1998) define IE as an activity of automatically extracting pre-specified sorts of information from short, natural language Texts, with the aim of populating a structured database. Various forms of information can be extracted from Incident Reports, they include entities, entity attributes (e.g. title of a person or type of organization), facts (e.g. relations between entities such as the company a person works for) and Events. In this study we focus on the extraction of entity relational facts.

The task of relation extraction from unstructured texts was first formulated as part of the Message Understanding Conference 1998 (MUC-7) ((NIST), 1998). (ACE) (2005) defined a relation as an ordered pair and stated that the goal of the relation task was to detect and characterize relations of targeted types between entities. Banko et al. (2007) define Relation Extraction (RE) as the task of recognizing the assertion of a particular relationship between two or more entities in text. To achieve this goal various approaches to relation extraction have been adopted, we review some of these approaches.

2.3 Entity Relational Facts

(ACE) (2005) defines Relations as ordered pairs of entities. This means that for one to identify a relation the sequencing of arguments therein is important. To achieve the ordering of entities in a relation two different argument slots (arg1 and arg2) are used for each relation e.g. arg1 <Relation> arg2. Further, it is important to note that relations unlike entities and events have no actual anchor in text and hence the need to delimit the relation extraction problem to relations expressed in a single sentence((ACE), 2005). Table 1 provides a summary of some likely Entity Relational facts and their corresponding arguments

Table 1: Entity Relations tagging guidelines (Source ACE 2005)

Relation Type	Relation Sub Type	Argument1	Argument2	Description
Physical	Physical.Located	Person	Facility, Location, Geo-Political Entity	Located Relation is re- mentions of E
	Physical.Near	Person, Facility, Geo-Political Entity, Location	Facility, Geo-Political Entity, Location	Indicates that entity is a par
Part-Whole	Part-Whole.Geographical	Facility, Location, Geo-Political Entity	Facility, Location, Geo-Political Entity	Indicates that entity is a par
	Part-Whole.Subsidiary	Organization	Organization, Geo-Political Entity	Captures the relationships GPEs.
	Part-Whole.Artifact	Weapon	Weapon	Characterizes objects and t type. This F Weapons.
Vehicle		Vehicle		
Personal-Social	Per-Social.Business Per-Social.Family	Person	Person	Personal-Soc The relation c
Org-Affiliation	Org-Aff.Employment	Person	Organization, Geo-Political Entity	Employment employers
	Org-Aff.Membership	Person, Organization, Geo-Political Entity	Organization	Membership organization

2.4 Approaches to Relation Extraction

Approaches to relational extraction can be broadly categorized into: Supervised, Semi-supervised and Unsupervised. In this section we examine each category and sample techniques in each.

2.4.1 Supervised approach to Relation Extraction

In this approach the relation extraction problem is formulated as a classification task {Bach, 2007 #31}. Whereby, a training set of negative and positive examples is used to train a classifier based on certain features.

Given a sentence $S = w_1 w_2 \dots e_1 \dots w_i \dots e_2 \dots w_{n-1} w_n$.

Where e_1 and e_2 are entities, a mapping function $f(.)$ can be given as:

$$F_R(T(S)) = \begin{cases} +ve & \text{if } e_1 \text{ and } e_2 \text{ are related by } R \\ -ve & \text{if otherwise} \end{cases}$$

$F_R(.)$ can be a discriminative classifier e.g. SVM, Voted Perceptron, Log-linear or it can be a multi class classifier

$T(S)$ can be a set of features extracted from the sentence

There are two main approaches that can be used to train a classifier for Relation Extraction: Feature based and Kernel based. Kudenko and Hirsh (1999) describe Feature-based learning algorithms as those that require the input data to be represented in a feature-vector format i.e. as a collection of feature value pairs, whereby a feature is a mapping function from the set of examples to a set of feature values (the feature value domain). For example the weather at a certain time point could be represented as a feature-vector using three features temperature (T), humidity (H) and pressure (P). The feature value domain for T and H can be represented as integers, while the feature value domain for P can be represented as the set flow, middle, high. Thus an example could be represented as the feature-vector ((T 50) (H 65) (P low)).

Moncecchi et al. (2003) argue that the problem with feature-based methods is that some natural language sentences cannot be easily represented with explicit feature vectors, in such cases feature extraction is a very complex task with very high dimensional vectors which create computational problems. To overcome this problem Kernel-based methods are used, these methods compute a similarity function (or kernel) between examples and discriminative methods are used to label new examples.

A kernel function over an object space X is a binary function

$$K: X * X \rightarrow [0; 1]$$

This function assigns a similarity score between two instances of X .

2.4.2 Semi Supervised Approaches to Relation Extraction

2.4.2.1 Dual Iterative Pattern Expansion (DIPRE)

Brin (1998) proposed DIPRE as a method of relation extraction which required minimum human intervention and used a small seed set of five (5) relations of the form (author, title), to find occurrences of all the seed sets books on the web. From the occurrences of these books patterns of the citations were induced and used to find further occurrences of the new patterns and to generate more patterns iteratively.

DIPRE works as follows:

1. Start with a small sample of the target relation R^1 i.e. Start with a few examples of a relationship e.g. (Shakespeare, Hamlet) for author book relation.
2. Find all occurrences of tuples of R^1 in D (Document collection) e.g. Shakespeare's works such as Hamlet. This can be represented as $O \leftarrow \text{FindOccurrences}(R^1, D)$
3. Generate patterns by generalizing the set of occurrences. $P \leftarrow \text{GenPatterns}(O)$
4. Search the database for tuples matching any of the patterns, use the patterns to extract more examples $R^1 \leftarrow M_D(P)$ note $(M_D(P))$ is the set of tuples that match P in D .
5. If R^1 is large enough return. Else go to step 2.

A DIPRE pattern is defined as a five-tuple: (*order, urlprefix, prefix, middle, suffix*) where *order* is a Boolean value indicating which entity in the relation occurred first, *urlprefix* refers to the web address of the information and *prefix, middle* and *suffix* are the strings that occur before the first entity, between the two entities and after the second entity.

2.4.2.2 Snowball

Agichtein and Gravano (2000) made improvements to DIPRE by developing a way to represent extraction patterns that would enable capturing of many valid tuples in a text collection. As part of Snowball a named entity tagger was included, this ensured that Snowball's patterns included named-entity tags. An example of a Snowball pattern is `<LOCATION>-based<ORGANIZATION>`. With this pattern not all string pairs connected by "-based" will be matched instead, `<LOCATION>` will only match a string identified by a tagger as an entity of type *LOCATION* and `<ORGANIZATION>` will only match a string identified by a tagger as an entity of type *ORGANIZATION*. This minimized the possibility of incorrectly identifying a relationship pair. For example consider the following sentences.

Joe Doe works on the ACME Poverty Eradication Program.

Tech Computers works on the Computer-In-School project.

Only the first sentence indicates an employer-employee relationship, the only way to know this is to recognize "Joe Doe" as a person's name, while "Tech Computers" is an Organization name.

2.4.3 Unsupervised Approaches to Relation Extraction

2.4.3.1 KnowItAll

KnowItAll was the first system to carry out unsupervised, domain independent, large-scale extraction from web pages. It achieved this by learning how to label its own training examples using a small set of domain independent extraction patterns (Etzioni et al., 2006). Instead of utilizing hand tagged training data, the system selects and labels its own training examples, and iteratively bootstraps its learning process. Etzioni et al. (2006) further argues that KnowItAll is relation-specific and requires a laborious bootstrapping process for each relation of interest, with a human user naming the relations of interest in advance.

2.4.3.2 Open Information Extraction

Banko et al. (2007) were the first to propose a single learning model of how relationships are expressed in a particular language. Open Information Extraction, was a shift from the traditional relational extraction task that required one to build distinct extractors for each relation of interest. To fulfill this shift the relation extraction problem was recast from a classification task, which required extractors to learn how to identify relation instances using surrounding context to an Open IE paradigm centered on identifying relational phrases i.e. phrases that denote relations in English. This approach ensured that:

- i.) The labor of building an Open IE system was independent of the number of target relations.
- ii.) There was a way for domain independent discovery of relations.

To prove the possibility of a single learning model, Banko (2009) studied how binary relations are expressed in English sentences and showed that 95% of relationships are consistently expressed using a compact set of relation-independent lexico-syntactic patterns. The patterns were grouped into the categories shown in the table below.

Relative Frequency	Category	Simplified Lexico-Syntactic Pattern	Example
37.8	Verb	E ₁ Verb E ₂	X created Y
22.8	Noun + Prep	E ₁ NP Prep E ₂	X is birthplace of Y
16.0	Verb + Prep	E ₁ Verb Prep E ₂	X moved to Y
9.4	Infinitive	E ₁ to Verb E ₂	X plans to acquire Y
5.2	Modifier	E ₁ Verb E ₂ Noun	X is Y winner
1.8	Coordinate _n	E ₁ (and , - :) E ₂ NP	X-Y deal
1.0	Coordinate _v	E ₁ (and ,) E ₂ Verb	X , Y merge
0.8	Appositive	E ₁ NP (: ,)?E ₂	X hometown : Y

Table 2: Taxonomy of Binary Relationships

Nearly 95% of 500 binary extractions were described using one of eight lexico-syntactic patterns. NP refers to noun phrases, E_i refers to entities, and Prep indicates a preposition. Source (Banko 2009 pg 12).

Banko (2009) argued that the above results only lent support to the possibility of open extraction and that simply applying the patterns was not a sufficient solution in itself due to concerns about precision and recall.

Using this paradigm Banko (2009) , Etzioni et al. (2006) developed TextRunner, a system that was able extract information from each sentence it encountered rather than require relationships be specified in advance. TextTrunner’s extractor module reads sentences and extracts textual tuples that aim to capture relationships e.g. given the sentence “Berkeley hired Robert Oppenheimer to create a new school of theoretical physics”, the extractor forms the triple (Berkeley, hired, Robert Oppenheimer). The triple consists of three strings where the first and third are meant to denote entities and the intermediate string is meant to denote the relationship between them.

Fader et al. (2011) identified two significant problems with Open IE as presented by Banko (2009) and the subsequent implementation of TextRunner: incoherent extractions and uninformative extractions. Incoherent extractions referred to cases whereby the extracted relation phrase lacked meaningful interpretation as a result of the learned extractor making a sequence of decisions on whether to include each word phrase. Uninformative extractions on the hand occur when extractions omit critical information. Fader et al. (2011) gives the following example; consider the sentence “ Hamas claimed responsibility for the Gaza attack”. Previous Open IE systems return the uninformative: (Hamas, claimed, responsibility) instead of (Hamas, claimed responsibility for, the Gaza attack).

To overcome these shortcomings Fader et al. (2011) proposed the use of a syntactic constraint to eliminate incoherent extractions, and to reduce uninformative extractions by capturing relation phrases expressed via light verb constructions. The syntactic constraint requires relation phrase to match the POS tag pattern shown

V V P VW*P
V = verb particle? adv?
W = (noun adj adv pron det)
P = (prep particle inf. marker)

Table 3: syntactic constraint relation phrase POS tag pattern.

Source Fader et al. (2011)

Fader et al. (2011) argued that the pattern limits relation phrases to be either a simple verb phrase (e.g., invented), a verb phrase followed immediately by a preposition or particle (e.g., located in), or a verb phrase followed by a simple noun phrase and ending in a preposition or particle (e.g., has atomic weight of). If there are multiple possible matches in a sentence for a single verb, the longest possible match is chosen.

Additionally Fader et al. (2011) proposed the use of a lexical constraint to weed out phrases that satisfy the syntactic constraint but are not useful relations in themselves. The lexical constraint separates valid relation phrases from over-specified relations; this is based on the intuition that a valid relation phrase should take many distinct arguments in a large corpus.

Fader et al. (2011) implemented the constraints in ReVerb Open IE System; this doubled the area under the precision-recall curve relative to TextRunner and ensured more than 30% of ReVerb’s

extractions were at a precision of 0.8 or higher. Though ReVerb was designed for Web-scale information extraction an executable jar file can be downloaded from <http://reverb.cs.washington.edu/reverb-latest.jar> which can be used to extract relations from smaller corpuses. ReVerb takes plain text or HTML as input, and outputs a tab-separated table of output. Each row in the output represents a single extracted (argument1, relation phrase, argument2) triple, plus metadata. The output has the following fields/columns:

S/No	Field Description
1	The filename (or stdin if the source is standard input)
2	The sentence number this extraction came from.
3	Argument1 words, space separated
4	Relation phrase words, space separated
5	Argument2 words, space separated
6	The start index of argument1 in the sentence. For example, if the value is i, then the first word of argument1 is the i-1th word in the sentence.
7	The end index of argument1 in the sentence. For example, if the value is j, then the last word of argument1 is the jth word in the sentence.
8	The start index of relation phrase.
9	The end index of relation phrase.
10	The start index of argument2.
11	The end index of argument2.
12	The confidence that this extraction is correct. The higher the number, the more trustworthy this extraction is.
13	The words of the sentence this extraction came from, space-separated.
14	The part-of-speech tags for the sentence words, space-separated.
15	The chunk tags for the sentence words, space separated. These represent a shallow parse of the sentence.
16.	A normalized version of arg1.
17.	A normalized version of rel.
18	A normalized version of arg2.

Table 4: ReVerb Output

source: <http://reverb.cs.washington.edu/ReadMe.html>

2.5 Comparative Analysis of alternative approaches to Relation

	Supervised	Semi Supervised
<i>What does extracting a new relation entail?</i>	Since relations are specified in advance. One has to find text documents with the relation mentions annotate and train the model to extract the relation from unseen texts. This needs a significant amount of labor.	Since relations are specified in advance one has to find labeled instances of the relation in the seed set to train the model. This needs a significant amount of labor.
<i>What does porting the system to a new domain require?</i>	Training the system with the set of relations to be found in the new domain	Introduce the new relations of interest to the seed set. This needs a significant amount of labor.
<i>What time does it take to extract relations?</i>	O(RD) D documents, R relations	O(RD) D documents, R relations
<i>Which approach is used in Relation Extraction task?</i>	Classification problem i.e. whether a relation is valid or not valid.	Bootstrapping
<i>What NLP tools are used?</i>	Textual analysis such: POS tagging, shallow parsing, dependency parsing is a pre-requisite.	DIPRE does not use NLP tools while Snobol Named Entity tagger uses NLP tools.
<i>What are the Inputs to the system?</i>	1. A target relation (e.g. Organizations and their locations) is provided to the system. 2. Hand-crafted extraction patterns or positive and negative instances of the relation.	Labeled instances of the relation
<i>What is the cost of development?</i>	O(R), R relations	

Table 5: Comparative Analysis of alternative approaches to Relation Extraction

2.6 Relational Representation Using Graphs

In the previous section, various approaches to relational extraction were identified; we now focus on how to handle the extracted entity relational facts. We identify and review the use of Graph data structures in representing entity relations and the use of Graph algorithms in identifying how two named entities are related in a text corpus. Grama et al. (2003) points out that graph theory provides a way to model many problems in computer science, and when these problems are expressed in terms of graphs they can be solved using standard graph algorithms.

Minkov (2008) argues that a graph schema naturally represents relational data, where nodes denote entities and directed typed edges represent the relations between them. Such graphs, he adds, are heterogeneous in the sense that they describe different types of objects and multiple types of links.

Formally, a graph $G = \langle V, E \rangle$ consists of a set of nodes V , and a set of labeled directed edges E . Graphs are used to model pair wise relations between objects from a certain collection. There are two standard ways to represent a graph in a computer program, either as a matrix or as a linked list (Grama et al. (2003)). The nature of a graph i.e. whether the graph is sparse or dense, determines which representation should be used. A graph $G = (V, E)$ is *sparse* if $|E|$ is much smaller than $O(|V|^2)$; otherwise it is *dense*. The matrix representation is useful for dense graphs and the adjacency list representation is more efficient for sparse graphs (Grama et al. (2003)).

Minkov (2008) defines a walk in a graph G as a sequence of nodes (n_1, n_2, \dots, n_i) such that each adjacent pair $(n_1; n_2), (n_2; n_3) \dots (n_{i-1}, n_i)$ are arcs in G . A path can be defined as a walk with no repeated nodes. If there is a path between two nodes u and v then we say that u is reachable from v . A graph walk can be used to extract nodes in the graph that are similar by virtue of their connectivity to the start nodes, this notion of similarity is often task dependent (Minkov (2008)).

Minkov (2008) uses an email corpus containing email messages and meeting entries, that are viewed as objects and represented using a graph. Additionally other entities were corresponded to objects of types; *messages, terms, email addresses, persons* and *dates*. Directed graph edges were used to represent relations such as *sent-from, sent-to* and *on-date*. The corresponding graph *schema* is detailed in the table below.

Source type	Edge type	Target type
<i>Message</i>	sent-from	<i>Person</i>
	sent-from-email	<i>email-address</i>
	sent-to	<i>Person</i>
	sent-to-email	<i>email-address</i>
	on-date	<i>Date</i>
	has-subject-term	<i>Term</i>
	has-term	<i>Term</i>
<i>Meeting</i>	Attendee	<i>Person</i>
	attendee-email	<i>email-address</i>
	mtg-on-date	<i>Date</i>
	mtg-has-term	<i>Term</i>
<i>Person</i>	sent-from ⁻¹	<i>Message</i>
	sent-to ⁻¹	<i>Message</i>
	attendee ⁻¹	<i>Meeting</i>
	alias	<i>email-address</i>
	as-term	<i>Term</i>
<i>email-address</i>	sent-to-email ⁻¹	<i>Message</i>
	sent-from-email ⁻¹	<i>Message</i>
	attendee-email ⁻¹	<i>Meeting</i>
	Alias ⁻¹	<i>Person</i>
	is-email ⁻¹	<i>Term</i>
<i>Term</i>	has-subject-term ⁻¹	<i>Message</i>
	has-term ⁻¹	<i>Message</i>
	mtg-has-term ⁻¹	<i>Meeting</i>
	is-email	<i>email-address</i>
	as-term ⁻¹	<i>Person</i>
<i>Date</i>	on-date ⁻¹	<i>Message</i>
	mtg-on-date ⁻¹	<i>Meeting</i>

Table 6: Email and meetings node and relation types

(Inverse edge types are denoted by a superscript.) Source (Minkov, 2008 pg 69)

As an example of how a sub graph can be represented from an email corpus consider the figure below; m_1 , m_2 and m_3 represent email messages, p_1 , p_2 and p_3 represents email addresses that a message could be sent to or sent from, t_1 , t_2 and t_3 represents terms that were contained in the subject line of the email message.

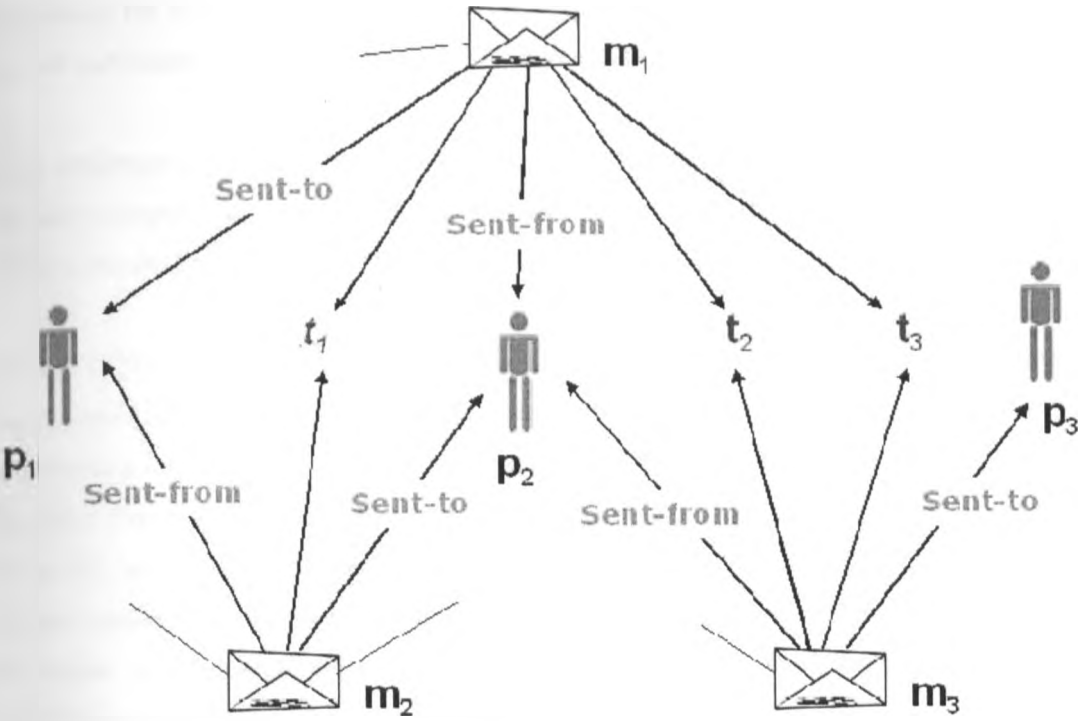


Figure 1: An example sub-graph, showing the connecting paths.

Source Minkov (2008, pg 51)

2.7 Related work

Various works exist in the context of the two sub problems identified of Relational Extraction and Relation representation. We reviewed the work of Freitag (1998), Banko (2009) and Fader et al. (2011) with a view of contextualizing our research and identifying an appropriate way of formulating the relation extraction task. We further reviewed the work of Minkov (2008) with a view of understanding how to effectively structure relational data as a graph.

A key challenge faced in solving the Relation Extraction sub problem is how to effectively build a domain independent relation extractor, bearing in mind the varied nature of the target audience of this research (Security, Insurance and Media).

Freitag (1998) used supervised learning techniques to build a domain independent extractor, which consisted of a package of machine learning techniques. He conducted experiments using four learners (a rote learner (Rote), a statistical term-space learner based on the Naïve Bayes algorithm (BayesIDF), a hybrid of BayesIDF and the grammatical inference algorithm Alergia (BayesGI), and a relational learner (SRV)) on three different document collections—electronic seminar announcements, newswire articles describing corporate acquisitions, and the home pages of courses and research projects at four large computer science departments. Three key contributions that arose from Freitag's work are:

First, Effective information extraction is possible without recourse to natural language processing. In domains where semantic and syntactic information is unavailable or difficult to obtain such as the web it is still possible to extract meaningful information. This view was reinforced by Brin (1998) use of DIPRE to extract (author book) relations from the web.

Second, there is no single learning approach that is suitable for all information extraction problems. Freitag (1998) highlighted the possibility of developing and using different learners to suit different document views to achieve optimal performance. The document views were:

- i. *Terms view*, regards a document as a sequence of terms. The bag-of-words model, which ignores ordering, is basically a weakening of this view
- ii. *Linguistic view*, each term belongs to a particular part of speech (Noun, pronoun, adjective etc)
- iii. *Typographic view*, each term can viewed as belonging to different sets such as numeric, punctuation, upper case etc

- iv. *Markup view*, from a markup point of view there may be meta-terms which provide role information about terms, HTML contains explicit meta-terms but even ASCII contains 'control' characters such as tabs and carriage return, the purpose of which is to partition terms
- v. *Layout view* can be regarded as an interpretation of the markup view by some application. Many important textual objects can be discerned only at this level such as paragraphs, headlines, tables, mail headers, signatures etc

The third contribution by Freitag (1998), is captured by the view that by combining trained information extractors one can realize substantial improvements over the performance of the best individual extractor.

Banko (2009) chose to use Natural Language Processing approach to the Relational Information Extraction problem. This approach resulted in a single learning model that was domain independent and language dependent, the model was implemented in Java as TextRunner, and was used for web scale relation extraction. Fader et al. (2011) identified shortcomings of TextRunner and proposed the use of the lexical and syntactic constraints to improve its performance. The improvements were implemented as REVERB; a web scale relation extraction system with a downloadable version for use on standalone machines.

For the relation representation sub problem we studied the work of Minkov (2008) who used graphs in two domains the Personal Information Management (PIM) domain and the processing of parsed text domain. In the PIM domain Minkov (2008) showed that email data, meeting entries and entities such as persons, dates, email addresses can be represented as a graph and a graph walk over this network naturally integrates textual and non-textual objects i.e. combining text, recipient information and a timeline. In the parsed text domain word mentions are represented as nodes and the syntactic structure, which binds these words as labeled edges denoting inter-word relations; graph walks are then applied to derive an extended measure of similarity, or relatedness between words.

Our research distinguishes itself from Minkov's work in that Minkov (2008) focuses on semi structured data in the PIM and parsed text domains whereas we focus on the use of graphs to represent entity relational facts in unstructured Incident Reports corpuses. Additionally we use an Open Information extractor; REVERB, to obtain the relational facts, whereas Minkov's work did not involve the use of a relational extractor. Finally, we locate our work as an extension of part of Minkov (2008) work using the knowledge and tools provided in the works of Banko (2009) and Fader et al. (2011).

2.8 Conceptual Framework

There are informal ways of viewing an unstructured Incident Report text corpus such as 'Subject view' whereby users see a corpus in terms of the topics or subjects contained therein; the 'Chronological view' whereby the Incident Report corpus is viewed as a series of documents arranged by the date they were written or the date the incidents contained therein occurred. Central to these views is the individual document, which is seen as whole and not as a sum of its constituent facts.

For us to solve the problem of finding out how two named entities are related to each other, we extend the work of Freitag (1998) by introducing an abstract view of an incident report – "Fact view". In this view each Incident Report is seen as a sum of its constituent facts. This is premised on the knowledge that individual Incident Reports contain a handful of facts that may be interrelated with other facts contained in other reports. The "Fact view" enables us to characterize a document corpus not as a collection of documents but as a collection of interlinked and/or interrelated facts.

From the above argument we can therefore conceptually model the problem as follows:

An organization's Incident Report corpus (D) can be regarded as a set of individual documents (d_i). This is represented as:

$$D = \{d_1, d_2 \dots d_n\}$$

Each document d_i in the set $\{d_1, d_2 \dots d_n\}$ contains facts $\{f_1, f_2 \dots f_k\}$ we call this set F^i . Each fact f_i in the set F^i is a relational fact of the form $E_1 <relation> E_2$

All facts that can be extracted from a corpus (D) can be represented as a set of facts (F). This can be expressed as:

$$F = \{f_1, f_2, f_3 \dots f_n\}$$

Therefore we can say that F^i is a subset of facts F which reside in a document d_i which is part of the document collection D

Therefore we can say

$$F^i \in d_i \in D$$

(Facts F^i exist in a document d_i which in turn exists in the document collection D)

From the foregoing we can formulate our problem as shown below:

We formulate the relation extraction problem as:

Given: Document collection $D = \{d_1, d_2, \dots, d_n\}$ and Named Entities $E = \{E_1, E_2, \dots, E_n\}$

Input: A set of English relational patterns used in Open Information Extraction paradigm (refer to Table 2)

Output: Fact collection $F = \{f_1, f_2, f_3, \dots, f_n\}$ such that f is a relational fact of the form $E_1 <relation> E_2$.

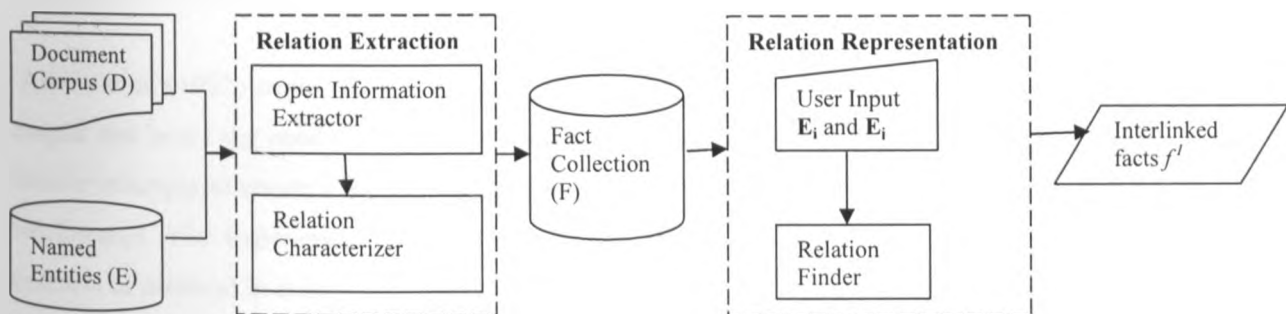
We formulate the problem of finding out how two named entities are related as:

Given: Fact collection $F = \{f_1, f_2, f_3, \dots, f_n\}$ over a document collection $D = \{d_1, d_2, \dots, d_n\}$ and Named Entities $E = \{E_1, E_2, E_3, \dots, E_n\}$

Input: Two named Entities that you wish to find out how they are related E_i and E_j

Output: $f^i = \{f_1, f_2, f_3, \dots, f_m\}$ such that the set f^i contains relational facts interlinking the relation between the two entities

Figure 2: Diagrammatic Representation of the Conceptual Framework



CHAPTER THREE: METHODOLOGY

3.1 Research Design

Cunningham (2000) argues that all natural languages share the great advantage of being the most expressive communication method available, and the great disadvantage of being the most expressive communication method available, and therefore inherently ambiguous since it is difficult to impose structure or stick to conventions. This ensures that we cannot exhaustively describe natural language and hence the need to sample it in order to achieve balance and representativeness that match the research question (McEnery et al., 2006).

Representativeness refers to the extent to which a sample includes the full range of variability in a population. McEnery et al. (2006) point out that for one to obtain a representative sample from a population the first concern to be addressed is how to define a sampling unit and the boundaries of the population e.g. for written text a sampling unit may be a book, periodical or newspaper; the population is the assembly of all sampling units while the list of sampling units is referred to as a sample frame. Biber (1993) argues that though researchers focus on sample size as the most important consideration in achieving representativeness (how many texts must be included in the corpus, and how many words per text sample), sample size is not the most important consideration in selecting a representative sample; rather, a thorough definition of the target population and decisions concerning the method of sampling.

Balance on the other hand refers to the proportional sampling from the target population of texts covering a wide variety of frequent and important text categories, this helps in achieving representativeness (McEnery et al. (2006)).

Atkins et al. (1992) argue that it is theoretically suspect to aim at achieving a perfectly 'balanced' corpus and hence the need to adopt a method of successive approximations; Whereby, the corpus builder attempts to create a representative corpus, which is analyzed to identify its strengths and weaknesses. The experience and feedback is used to enhance the corpus by the addition or deletion of material in continually repeated cycle.

This Research Design outlines: the Sampling procedure used to create a model Test corpus, taking into consideration the arguments made by Cunningham (2000), McEnery et al. (2006), Biber (1993) and Atkins et al. (1992); the Observation procedure used to annotate the documents in the Test corpus; the Statistical procedure used to determine how many observations were made and the Experimental procedure used to conduct experiments.

3.1.1 Sampling Procedure

In this part of the research, we identified and organized written texts that fit the description of Incident Reports with a view of creating two (2) model text corpuses (a Development and Test corpus) for use by the prototype application. The target population being news stories published in newspapers and websites describing incidents that had occurred. A purposive sampling procedure was adopted, where news stories selected for inclusion in the model text corpuses fulfilled the following requisites:

1. Published by a Kenyan media house in a newspaper or website between 1st February 2012 and 15th June 2012.
2. The news story referred to an incident that occurred within Kenya.
3. The story contained at least one wh-pattern i.e. at least one mention of a Location, Person or Facility.
4. The story had minimum length of three (3) sentences and maximum length of thirty (30) sentences.
5. The story was written in formal English with the purpose of informing.

The end result of this stage was two model text corpuses of ten (10) news stories each. One of the text corpuses –Development Corpus was used in the development process of the prototype application, whereas the –Test Corpus was used to test the developed prototype.

3.1.2 Observation Procedure

Here we annotated the sampled texts in order to come up with a standard that provided a basis for measuring the performance of the developed prototype.

Cunningham et al. (2010) argues that when we evaluate the performance of a processing resource such as tokeniser, POS tagger, or a whole application, we usually have a human-authored ‘gold standard’ against which to compare our software. However, it is not always easy or obvious what this gold standard should be, since different people may have different opinions about what is correct. To solve this problem we used two human annotators, and compare their annotations by calculating the Inter-Annotator Agreement (IAA).

For us to create a ‘gold standard’ upon which the performance of the prototype was measured, two human annotators separately read the model Test corpus and annotated all entity relational facts they could find according to the following relational specification guidelines.

1. The relationship could have held at any point in time past, present or future.
2. Speculation on a relationship was annotated positively e.g. *Joshua may travel to London.*
3. The relationship must be stated within the sentence in question and should not be inferred from other information.
4. Only a relationship between two entities indicated and not repeated mentions of the same entity in a sentence were considered.

The Test corpus (without annotation) was used as input to the developed system and the number of entity relational facts extracted was counted and tabulated against the results of the human annotated texts.

3.1.3 Statistical Procedure

In this section we determined how many observations were to be made and how the analysis was to be conducted.

The number of observations was guided by the identified news stories included in the Test corpus which had a minimum of three (3) sentences each and a maximum of thirty (30) sentences. The number of observations was therefore dependent upon the number of entity relational facts that could be extracted from individual sentences in the news stories.

The results of the annotations by the two annotators A and B were tabulated in a table of the form.

File Name: test.txt			
Sentence	Annotator		Agree
	A	B	
The rangers led by Warden <i>Joshua ole Naiguran</i> , arrested the poachers and <u>recovered</u> an <i>AK47 and G3</i> rifles and 66 rounds of ammunition.	Yes	No	False

Table 7: Summary of Annotations by A & B

To calculate IAA the following notation will be adopted

A_o . . . observed (or “percentage”) agreement

A_c . . . expected agreement by chance

General form of chance-corrected agreement measure R:

$$R = \frac{A_o - A_e}{1 - A_e}$$

To ensure validity of the observations made by the annotators, the value of R had to be greater than 50% to ensure that the agreement by the annotators was not by chance. If R was less than 70% a new pair of annotators was identified and presented with the same model corpus, if the result of R is still less than 70% then news stories in the model corpus that had the highest disagreement between annotators were replaced and the model corpus resubmitted for annotation.

3.1.4 Experimental Procedure

In this section we outline how the techniques and procedures specified in the Sampling procedure, Observation procedure and Statistical Procedure sections are organized in order to conduct experiments using the following set of apparatus.

- i. The prototype application.
- ii. A list of Named Entities e.g. people names and Locations
- iii. Model Test Corpus

The following sequence of steps was followed in conducting the experiments.

- I. Run the prototype application, select the option of "*File Reader*" and specify the folder with the Test corpus as the Input folder e.g. C:\Corpus\Test. Specify an existing folder as the Output folder e.g. C:\Corpus\Output. Select the "Extract candidate Relations" button this step will extract candidate relations using ReVerb from the Test Corpus and store them in the specified output folder.
- II. Once the candidate relations are extracted, we click on the "*Tag and Characterize Relation*" button prototype menu.
- III. Review the Extracted Relations for correctness and similarity to the human annotated relations.
- IV. Review the suggested Named Entities and include the in the Named Entity Table.
- V. Repeat step II to IV after you add new named entities in the Named Entity Dictionary.

3.1.5 Justification for the Design

The research design was tailored to ensure validity of the results (i.e. we measure what was intended to be measured) and the reliability of measure (i.e. the same results will be obtained by another experimenter under the similar conditions on the same sample).

To ensure validity we adopted a purposive sampling procedure that, targeted news stories with characteristics outlined in the sampling procedure section. This ensured that we came up with a representative and relatively balanced corpus, from the sample frame of new stories published in local newspapers. Alternative sampling procedures such as random sampling were unsuitable for this task.

To ensure reliability, two (2) annotators (A and B) were provided with a set of guidelines (as outlined in the observation procedure section) to use in annotating the model Test corpus, their annotations were tabulated and corrected for chance agreement.

Additionally the choice of using news stories for the model text corpuses was motivated by the respect for copyright while still exploiting the stories similarity to the ideal Incident Reports.

3.1.6 Sources of data and relevance of data to the problem

Newspaper articles written in English covering a diverse set of incidents in Kenya were collected and used to model two (2) text corpuses for use in this research. First the articles were examined for professional authorship i.e. appropriate use of language, syntax and semantics.

Secondly, the articles were examined for relevance and suitability of them being classified as Incident Reports i.e. do they contain Entity Relational facts? This enabled us model text corpuses similar to those that may be found in a typical an organization within the Security or Insurance sectors.

The newspaper articles were annotated and used to develop a benchmark or baseline (gold standard) upon which the developed prototype was evaluated. The selected articles were the main input of the prototype system.

3.1.7 Procedures and methods for data collection.

News stories in the Test corpus were serialized and printed out and given to two annotators (A and B). For each story the annotators identified sentences that contained entity relational facts, double underlined entity names and single underlined relations contained therein. For example the sentence “*The rangers led by warden Joshua ole Naiguran, arrested the poachers and recovered an AK47 and G3 riffles and 66 rounds of ammunition.*” was annotated as follows “*The rangers led by warden Joshua ole Naiguran, arrested the poachers and recovered an AK47 and G3 riffles and 66 rounds of ammunition.*” Once the annotators completed the annotation the results were tabulated in a table as shown in Table 7. Additionally the sentences were captured and stored in a database table with their corresponding annotation.

Two independent annotators annotated the Test corpus, since different people may have different opinions of what is correct. This enabled us to calculate an Inter Annotator Agreement (IAA), which was used to determine the ceiling for the performance developed prototype.

Once the prototype application was run the extracted relation were tabulated side by side with human annotators (A&B) extractions as shown in the table 8 below.

File Name: test.txt					
Sentence	Annotator		Ann.	Prototype	Correct
	A	B	Agree	Extracted	Extraction
<u>Joshua Maina</u> , who was <u>travelling to Kisumu</u> .	Yes	Yes	Yes	Yes	Yes
The rangers led by Warden <u>Joshua ole Naiguran</u> , <u>arrested the poachers and recovered an AK47 and G3 riffles and 66 rounds of ammunition.</u>	Yes	No	No	Yes	No

Table 8: Summary of Human Annotated and Prototype Annotated Extractions

The total number of relations extracted by the human annotators (T_H) was counted by counting the number of Yes entries in the ‘Ann. Agree’ column. To find the total number of relations extracted by the developed prototype (T_P) we counted the number of Yes entries in the ‘Prototype Extracted’ column.

The entries in the ‘Correct Extraction’ column were derived from comparing the entries in the ‘Ann. Agree’ and ‘Prototype Extracted’ columns, the comparison was conservative e.g. in the sentence “*The rangers led by Warden Joshua ole Naiguran, arrested the poachers and recovered*

an AK47 and G3 rifles and 66 rounds of ammunition” the annotators identified the entity relation fact as: Joshua ole Naiguran recovered AK47 and G3 and the prototype application extracts Joshua ole Naiguran recovered AK47 the extraction was judged as correct.

The total number of Yes entries in the ‘Correct Extraction’ column was counted (T_C). In summary we can say:

- i. Total number of relations extracted by the human annotators = T_H
- ii. Total number of relations extracted by Prototype system = T_P
- iii. Total number of correct relations extracted = T_C

3.1.8 Data analysis methods and their justification

Precision measures the number of correctly identified entity relations as a percentage of the number of relations identified. In other words, it measures how many of the relations that the system identified were correct.

$$Precision = \frac{T_C}{T_P} \times 100$$

Error rate is the inverse of precision, and measures the number of incorrectly identified entity relations as a percentage of the relations identified. It is sometimes used as an alternative to precision.

$$Error\ rate = \frac{T_P - T_C}{T_P} \times 100$$

Recall measures the number of correctly identified relations as a percentage of the total number of correct relations. In other words, it measures how many of the relations that should have been identified actually were identified, regardless of how much spurious identification was made.

$$Recall = \frac{T_C}{T_H} \times 100$$

3.2 System Design and Implementation

To conduct the research as outlined in the Research Design section we developed computer programs to extract and characterize Entity Relational facts in a manner similar to what a human annotator would do. To facilitate the development of the computer programs two alternative methodologies were considered: Structured System Analysis and Design (SSADM) and Prototyping.

SSADM consists of five (5) phases; Feasibility Study, Requirements Analysis, Requirements Specification, Logical System Specification and Physical Design. It (SSADM) adopts the Waterfall model of systems development, where each phase has to be completed before subsequent phase can begin. The output of one phase forms the input of the next phase. SSADM provides for three (3) interdependent views; Logical view (used to identify, model and document data), Data Flow view (identify, model and document how data moves in an information system), Entity Model view (identify, model and document events that affect each entity and their sequence) of looking at a system.

Prototyping on the other hand involves the building of a working model of the candidate system for evaluation. There are two main approaches to prototyping; Throw away prototyping whereby prototype(s) are built within the user requirements analysis phase to elicit user requirements and abandoned when this phase is complete, and Evolutionary prototyping whereby the final system gradually evolves from a series of prototypes. The diagram below outlines the Evolutionary Prototyping process.

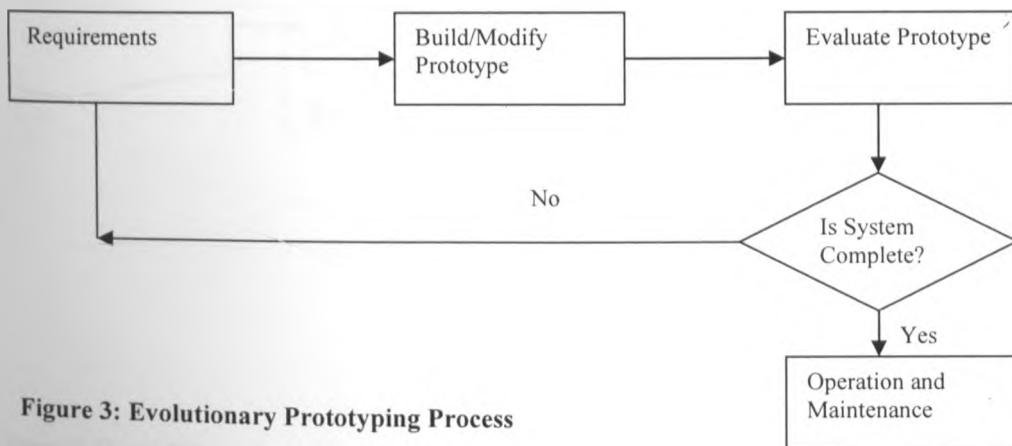


Figure 3: Evolutionary Prototyping Process

We adopted Evolutionary Prototyping over SSADM after considering the nature of the problem being tackled and the utilization of resources. In considering the nature of the problem being tackled we observed that Natural Language is inherently ambiguous leading to difficulties in imposing structure or sticking to conventions. In this case prototyping offers the advantage that

one need not obtain all the requirements before embarking on the development, this gave us the flexibility to refine and test new requirements as they arose for different test cases of the Development corpus Incident Reports that were being examined.

When considering the utilization of resources we observed that Evolutionary Prototyping ensures that resources are well utilized since each requirement change that results in the iteration of the prototype has a shorter turnaround time, unlike in SSADM.

3.2.1 System Architecture

Pressman (2001) defines the architecture of a system as a comprehensive framework that describes a system's form and structure i.e. its components and how they fit together. Pressman (2001) cites Bass et al. (1998) has having identified three (3) reasons why software architecture representations are important, the reasons are: First it enables communication between stakeholders, secondly it highlights early design decisions and thirdly it creates a relatively small intellectually graspable model of the system structure and interrelationship between components.

The pipeline system architecture that is widely used in developing natural language processing systems was adopted. In this architecture the output of one component of the architecture is used as the input of the succeeding component. The overall architecture of the prototype is as shown in figure 4 below.

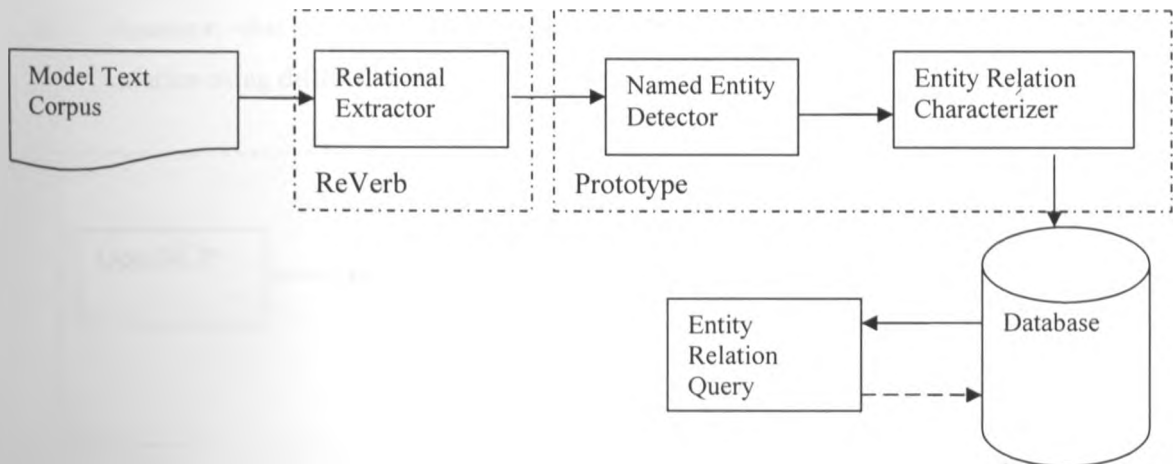


Figure 4: Overall System Architecture

3.2.2 Component Description

3.2.2.1 Model Text Corpus

The Model text corpuses (Test and Development corpus) that were created by sampling Natural Language texts as outlined in the Sampling Procedure (section 3.1.1) were used as the first component of the pipeline architecture. The model corpuses consisted of ten (10) news stories each, stored in a single folder in soft copy, in plain text format. The Development corpus was used during the development of the prototype application and the Test corpus was used during the evaluation of the prototype application.

3.2.2.2 Relational Extractor

The Relational Extractor component uses ReVerb a application developed by Fader et al. (2011) as a successor to TextRunner a web scale relational fact extractor. ReVerb consists of the following components.

- i. OpenNLP part of speech tagger: - this component takes in a natural language sentence tokenizes the sentence, tags the words with the respective parts of speech using the WordNet dictionary. Additionally, the tagger creates chunk tags using the IOB format.
- ii. Learner: - this component is trained using a small corpus and a set of relational independent heuristics its output is a single extraction model of English relationships outlined in the Relational Taxonomy shown in Table 2.
- iii. Extractor: -this component determines what constitutes a valid relation. It utilizes the syntactic constraint outlined in Table 3.
- iv. Assessor: -this component identifies instances describing the same real-world object or relation using different names.

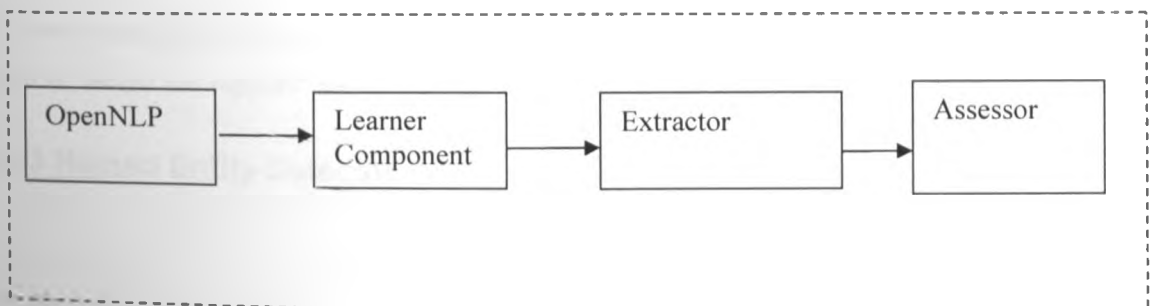


Figure 5: ReVerb Components

The Relational Extractor takes in one file at a time from the model text corpus; the file is processed by ReVerb which outputs candidate relations and the associated metadata. For example take the following sentence “Chama Cha Uzalendo yesterday said that it would not support Mr Uhuru Kenyatta 's presidential bid .” which was extracted from a news story

appearing in The Nation newspaper, of 17th March 2012. REVERB takes the sentence as input and gives the output shown in the table below.

S/No	Field Name	Output
1	Filename	C:\Nation-17-3-2012.txt
2	Sentence No.	1
3	Argument1	It
4	Relation phrase	would not support
5	Argument2	Mr Uhuru Kenyatta 's presidential bid
6	Argument1 start index	6
7	Argument1 end index	7
8	Relation phrase start index	7
9	Relation phrase end index.	10
10	Argument2 start index.	10
11	Argument2 end index	16
12	Extraction confidence	0.09972083123023254
13	Original sentence	Chama Cha Uzalendo yesterday said that it would not support Mr Uhuru Kenyatta 's presidential bid .
14	Part-of-speech tags for sentence	NNP NNP NNP NN VBD IN PRP MD RB VB NNP NNP NNP POS JJ NN .
15	Chunk tags for sentence.	B-NP I-NP I-NP B-NP B-VP B-SBAR B-NP B-VP I-VP I-VP B-NP I-NP I-NP I-NP I-NP I-NP O
16.	Normalized version of arg1.	It
17.	Normalized version of relation.	Support
18	Normalized version of arg2.	mr uhuru kenyatta 's presidential bid

Table 9: Sample Output of Relation Extractor

In the above case REVERB has identified the Entity relation fact whereby, Argument 1 is “It” the relation is “*would not support*” and Argument 2 “*Mr Uhuru Kenyatta's presidential bid*”.

3.2.2.3 Named Entity Detector (NED)

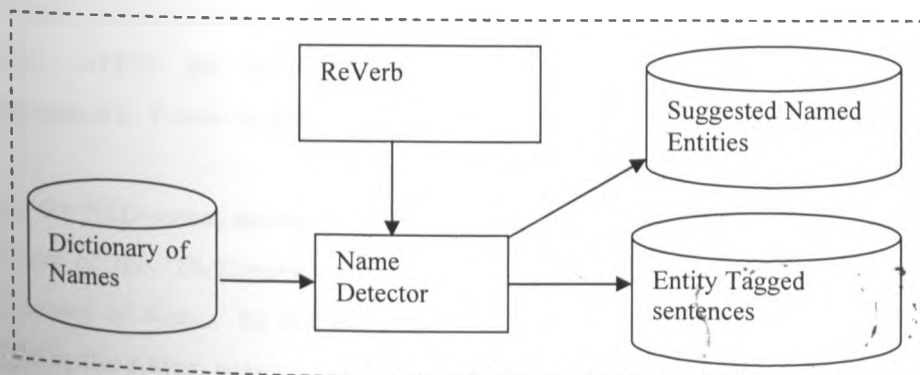


Figure 6: Structure of Named Entity Detector

The Named Entity Detector takes in the output of the Relational Extractor, it uses the IOB chunks in the 'Chunk tags' field (marked by serial No.15 in Table 9) to detect named entities using a greedy algorithm and a dictionary of entity names and their types. IOB tags are a standard way to represent chunk structures in files with the 'I' prefix being used to represent whether a phrase is 'IN' a chunk, the 'O' prefix used to represent whether a phrase is 'OUT' of a chunk and the 'B' prefix to represent the 'BEGINNING' of a chunk.

The output of the NED is an Entity tagged sentence e.g. take the sentence from the prior example with the corresponding Part-of-speech tags and Chunk tags shown in the table below.

Original sentence	Part-of-speech tags for sentence	Chunk tags for sentence.
Chama	NNP	B-NP
Cha	NNP	I-NP
Uzalendo	NNP	I-NP
Yesterday	NN	B-NP
Said	VBD	B-VP
That	IN	B-SBAR
It	PRP	B-NP
Would	MD	B-VP
Not	RB	I-VP
Support	VB	I-VP
Mr	NNP	B-NP
Uhuru	NNP	I-NP
Kenyatta 's	NNP POS	I-NP I-NP
presidential	JJ	I-NP
Bid	NN	I-NP
.	.	O

Table 10: Sample Sentence with Corresponding POS tags and Chunk Tags

When the NED takes the above as input, it outputs the tagged sentence shown below.

<ORG> Chama Cha Uzalendo </ORG> yesterday said that it <REL>would not support</REL> Mr <PER> Uhuru Kenyatta </PER> 's presidential bid . Entity tags <ORG>...</ORG> and <PER>...</PER> are inserted around parts of the text that the system recognizes as an Organization or Person respectively.

When the NED comes across a noun phrase that is not contained in the English dictionary and is not part of the 'Dictionary of Names', it suggests the noun phrase for inclusion into the 'Dictionary of Names' by including the noun phrase in a 'Suggested Named Entities' table for review by the system user.

Below is a pseudocode for implementing the Named Entity Detector.

Function NameEntityDectector (Sentence, ChunkList)

Intitalize counter,startIndex, endIndex to 0

WHILE counter < length of Chunklist

// This section finds the startIndex and endIndex of the Named Entity

IF ChunkList[counter] =='B-NP'

 SET startIndex = counter

IF ChunkList[counter] =='B-NP' and ChunkList[counter + 1] <>'I-NP'

 SET endIndex = counter

IF ChunkList[counter] =='B-NP'and ChunkList[counter + 1] =='I-NP'

 SET tempcounter = counter

WHILE tempcounter < length of Chunklist

 IF ChunkList[tempcounter] =='I-NP') and (ChunkList[tempcounter + 1] <>'I-NP'

 SET endIndex = tempcounter

 break

 Increment tempcounter by 1

END WHILE

// This section checks if the identified name is in Named Entity dictionary

IF ChunkList[counter] =='B-NP'

 initialize string, word to null

 SET tempcounter = startIndex

WHILE tempcounter < endIndex

 SET word = Sentence[tempcounter]

 IF word is in NEDdictionary

 SET str = str+ word tagged with tag NEDdictionary

 ELSE

 SET str = str+ Sentence[tempcounter]

 IF str in NEDdictionary

 SET str = str+ str tagged with tag NEDdictionary

 Increment tempcounter by 1

END WHILE

Increment Counter by 1

END WHILE

3.2.2.4 Entity Relation Characterizer

The Entity Relation Characterizer takes in the Entity tagged sentence that was output by the NED and determines whether the relation is valid. To determine if an entity tagged sentence is valid the Characterizer checks if the relation conforms to the form `<ENTITY>..</ENTITY><REL>...</REL> <ENTITY>..</ENTITY>`. Below is the pseudo code for implementing the Entity Relation Characterizer

Function characterizeRelation (fileName, sentenceNo, originalSentence, new_sentence)

Convert new_sentence to a list called RelationSentence

Initialize Counter, StartIndex, EndIndex = 0

*//This section identifies Entities and relations in a sentence and appends them to a list
//EntityRelationList for further processing in the next section.*

WHILE Counter < length(RelationSentence)

 IF RelationSentence [Counter] is a Opening Entity tag e.g. <PER>

 StartIndex = Counter

 IF RelationSentence [Counter] is a Close Entity tag e.g. </PER>

 EndIndex = Counter

 IF StartIndex <> EndIndex and EndIndex <> 0

 TempCounter = startIndex

 WHILE TempCounter < EndIndex

 Concatenate stringVariable with RelationSentence [TempCounter]

 Increment TempCounter by 1

 Append stringVariable extracted EntityRelationList []

 Initialize StartIndex, EndIndex = 0

 Intialize stringVariable to Null

 Increment Counter by 1

//We use the EntityRelationList to store the Entities and to determine the validity of Relations

Initialize TempCounter = 0

WHILE TempCounter < length (EntityRelationList)

 IF Entity EntityRelationList [TempCounter] is NOT contained in the InvertedIndex

 Append EntityRelationList [TempCounter] to InvertedIndex

 SET EntityIndex = Retrieve Entity Index of EntityRelationList [TempCounter]

 Increment TempCounter

IF Elements IN EntityRelationList conform to the pattern

`<ENTITY>..</ENTITY> <REL>..</REL> <ENTITY>..</ENTITY>`

Extract and store to ExtractedRelation Table

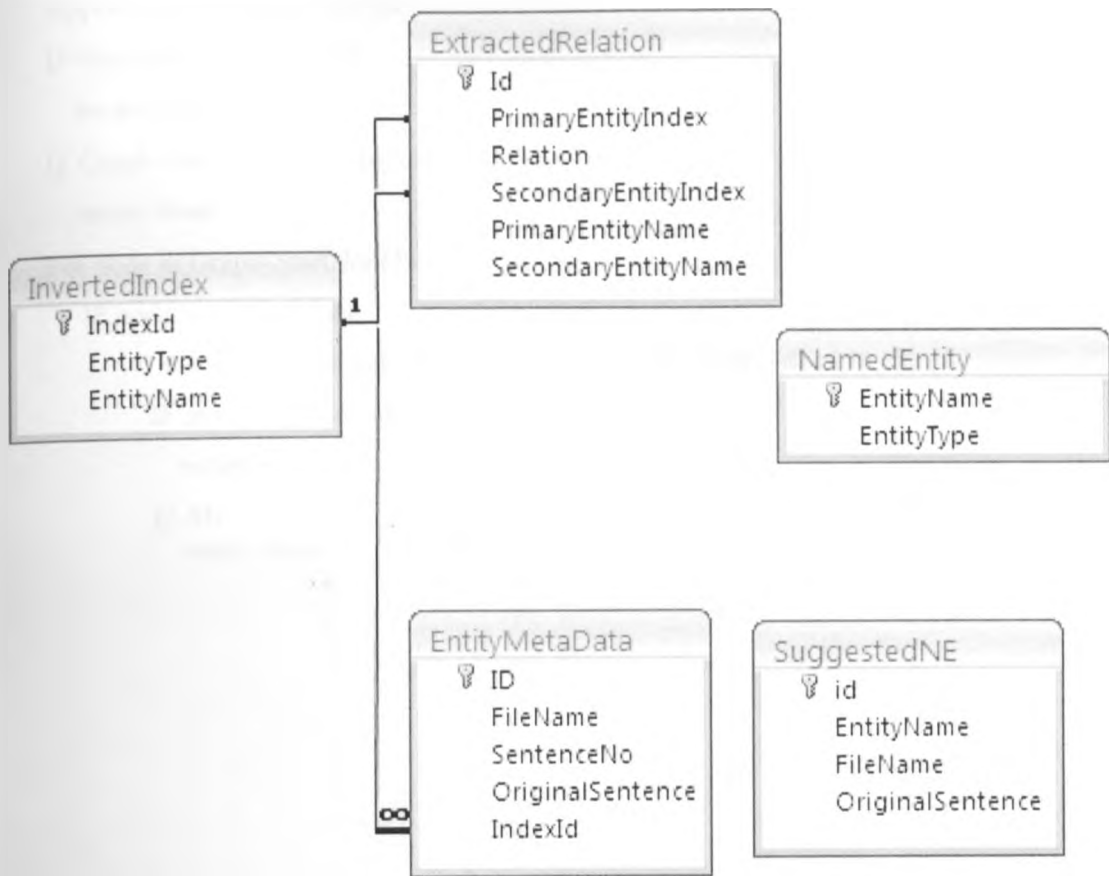
3.2.2.5 Database Structure

There are five (5) main tables used by the prototype application

TABLE NAME	TABLE DESCRIPTION
NamedEntity	This table stores a list of names and their types. It is used as a dictionary of what the system knows. The table consists of two fields <i>EntityName</i> and <i>EntityType</i> . <i>EntityName</i> is the Primary key.
InvertedIndex	Stores names of Entities that have been detected from the user's unstructured text corpus. The table consists of three fields <i>IndexId</i> , <i>EntityName</i> and <i>EntityType</i> . <i>IndexId</i> is the Primary key.
Extracted Relation	Stores <i>IndexId</i> 's of Entities and the relation between the Entities that have been extracted from a user's unstructured text corpus.
EntityMetaData	Stores the sentence that an Entity was extracted from. The table consists of five (5) fields; <i>ID</i> which is a unique identifier, <i>FileName</i> which is the name of the file that the sentence was extracted from, <i>SentenceNo</i> which is a numerical position of the sentence in the file that it was extracted from, <i>OriginalSentence</i> which is the sentence that contains the Entity that was extracted, <i>IndexId</i> which is a foreign key from the InvertedIndex Table.
SuggestedNE	Stores candidate Entities that the system did not tag as entities but which it considers may be likely Entities that a user can review for inclusion into NamedEntity Table.

Table 11: System Database Structure

Figure 7: System Database Structure



3.2.2.6 Entity Relation Query Engine

This component creates a graph data structure using data retrieved from the *EntityRelation* Table. It then uses a ‘transitive closure’ algorithm to find a path between two graph nodes, which represent Entities. Below is the pseudocode for implementing the Entity Relation Query Engine.

Function CreateGraph ()

//Create a Graph structure using the dictionary data structure

Retrieve all IndexId from InvertedIndex Table

For each IndexId retrieved

 SET TargetNodeList =Retrieve SecondaryEntityIndex from ExtractedRelation

 IF TargetNodeList is not empty

 MyGraph [Indexed] = TargetNodeList

Function FindPath (Graph, StartNode, EndNode, PathList)

Append StartNode to PathList

IF StartNode == EndNode

return PathList

IF Graph does not have StartNode

return None

For node in Graph[StartNode]:

IF node not in PathList

newpath = find_path(Graph, node, EndNode, PathList)

IF newpath is not empty

return newpath

ELSE

return None

3.2.3. System Test Plan

Test	Objective	Procedure
Unit Testing	To ensure that the Named Entity Detector, Entity Relation Characterizer and Relational Query Engine components function correctly as individual units	Use sample sentences drawn from Development corpus as test cases and apply testing techniques on the NED. Use the Output of the NED as input to the Relation Characterizer. Use Relational Query Engine to connect to the database.
Integration Testing	To ensure that the Relational Extractor can read the Model Text Corpus	Run relational extractor taking model text as Input
	To ensure that the Named Entity Detector can read the output of the Relational Extractor	Run the Named Entity Detector specifying the Relational Extractor output as the Input
	To ensure that the Relational Characterizer can read the output of the Named Entity Detector	Run the relational Characterizer specifying the output of the Named Entity Detector as Input
	To ensure that the Relational Characterizer can save its output on the database	Create an appropriate database connection through ODBC and Run the Relational Characterizer
	To ensure that the Relational Engine can receive user input and connect to the database	Provide two named Entities on the prototype interface form.

Table 12: System Test Plan

3.2.4 System Implementation

The Named Entity Detector, Entity Relation Characterizer and Entity Relation Query components of the prototype application were implemented in Python 2.6 which offers extensive libraries to support string processing. Python 2.6 for Windows is available at <http://www.python.org/ftp/python/2.6/python-2.6.msi>.

Further the Named Entity Detector and Entity Relation Characterizer additionally use NLTK a platform used for building Python programs that process human language. NLTK can be downloaded from <http://www.nltk.org/>. Additionally, pyAML a Python implementation of YAML which is a data serialization format designed for human readability and interaction was used, it can be downloaded from <http://pyyaml.org/download/pyyaml/PyYAML-3.10.win32-py2.6.exe>.

The Relational Extractor, is implemented in Java, it is an executable jar file that was downloaded from <http://reverb.cs.washington.edu/reverb-latest.jar>.

The Database component is implemented in Microsoft Access; the Entity Relation Characterizer and the Entity Relation Query Engine components connect to the Access database through an ODBC connection implemented by the pyodbc component downloaded from <http://pyodbc.googlecode.com/files/pyodbc-3.0.2.win32-py2.6.exe>.

Installation

1. Copy *reverb-latest.jar* file to C drive.
2. Install *python-2.6.msi* to the directory C:\Python26.
3. Install *nlk-2.0b9.win32.msi* to the folder C:\Python26.
4. Install *pyodbc-3.0.2.win32-py2.6.exe*
5. Install *pyYAML-3.10.win32-py2.6.exe*
6. Copy Microsoft Access file named *RelationExtraction.mdb* to the folder C:\Corpus.
7. Copy the Test Corpus folder to the folder C:\Corpus.
8. Create an ODBC connection named Test that points to the MS Access Database.
9. Copy the Files *RelationExtractor.py*, *RelationQueryEngine.py* and *TagAndCharacterize.py* to the folder C:\Python26.

To run the prototype double click on the MS Access file named *RelationExtraction.mdb*.

3.3 Limitations of Methodology

One of the key limitations of the methodology adopted is captured in the argument posed by Atkins et al. (1992) who argued that when building a Natural Language corpus, it is difficult to delimit the total population in a rigorous way; this means that given the sheer size of population (in our case incident news stories) and the available computation resources, it will always be possible for one to demonstrate that some feature of the population is not represented in the sample.

Another limitation is the absence of an obvious unit upon which to sample language and /or define a population. Sampling of a language can be based on words, sentences or texts among other things (Atkins et al., 1992) . To address this limitation we adopted a purposive sampling procedure, in the sampling frame of Kenyan newspapers and a sampling unit of a news article with characteristics of an Incident Report.

There are limitations that arise when annotation of a text corpus is done by a human being, key of which, is that different people have different language capabilities and the identification of correct relations in the model corpuses may be affected if the person(s) selected to identify entity relations is/are not well versed in the English language or their idiolect differs widely. Akmajian et al. (2001) defines idiolect as the language of a particular individual, and notes that the idiolect meaning of a word can differ from one person to another. To overcome this hurdle two (2) independent annotators with a good command of English were selected to annotate the model text corpus identifying the required entity relations.

Finally, in its current design the entity characterizer only considers entity relational facts of the form *<Entity> <Relation> <Entity>*. This does not cover lexico-syntactic patterns of the categories Coordinate_n and Coordinate_y outlined in Table 2, which have a relative frequency of 1.8 and 1.0 and whose sample sentences include X-Y deal or X, Y merger.

CHAPTER FOUR: RESULTS AND ANALYSIS

At the onset of the study we sought to answer the research question “What Key issues should be addressed in order to develop an Entity Relation characterizer?” To answer this question we identified four (4) key objectives:

1. To create a Test corpus for use by the Prototype application.
2. To design an entity relation characterizer.
3. To develop a prototype of the entity relation characterizer.
4. To evaluate the performance of the prototype on the Test corpus.

We outline the results of our research on the basis of the aforementioned objectives

4.1 Creation of a model Test corpus

For the first objective which was “To create a model Test Corpus”, ten (10) news stories of the characteristics outlined in the table below were selected as a representative sample.

File Name	No. of Sent.	Source	Author	Date
Nation-06-06-2012	15	http://www.nation.co.ke	Oliver Mathenge	6 th Jun 2012
Nation-18-04-2012	20	http://www.nation.co.ke	Fred Mukinda and Samuel Koech	18 th Apr 2012
Nation-17-03-2012	3	Nation Newspaper	-	17 th Mar 2012
Star-23-06-2012	5	The Star newspaper	Nzau Musau	23 rd June 2012
Nation-13-06-2012	7	http://www.nation.co.ke	Nation Correspondent	13 th Jun 2012
Standard-17-02-2012	10	Standard Newspaper	Leonard Korir	17 th Feb 2012
Star-02-03-2012	10	http://www.the-star.co.ke	Hussein Salesa	2 nd Mar 2012
Star-12-03-2012	6	http://www.the-star.co.ke	Kirimi Murithi	12 th Mar 2012
Star-27-03-2012	19	http://www.the-star.co.ke	Raphael Mwadime	27 th Mar 2012
Star-16-3-2012	13	http://www.the-star.co.ke	Mosoku Geoffrey	16 th Mar 2012

Table 13: Description of News stories included in Test Corpus

The news stories cover the following topics: Poaching, Politics and Accidents, and are written by a diverse set of authors to encompass variability in language use and to achieve Balance and Representativeness in the Test corpus. See Appendix 2 for the contents of the individual news stories.

4.2 Design of Entity Relation Characterizer

For the second objective which was “To design an entity relation characterizer”, the pipeline system architecture was adopted, which consisted of the following components: a text corpus, a relational extractor, a named entity detector, a relation characterizer, a database and a relational query engine. The output of one component served as the input of the succeeding component.

4.3 Prototype Development

For the third objective which was “To develop a prototype of the Entity Relation Characterizer”, three Python programs were developed *RelationExtractor.py*, *RelationQueryEngine.py* and *TagAndCharacterize.py* using Python programming language. Additionally a MS Access database *RelationExtraction.mdb* for use in conjunction with the programs created. See Appendix 3 for the program code.

4.4 Prototype Evaluation

For the fourth objective which was “To evaluate the performance of the prototype on the Test corpus”, Five (5) experiments were conducted with the prototype as outlined below. Experiment 1, 2, 3 and 4 aimed to extract and characterize relations as outlined in the succeeding sections. Experiment 5 sought to simulate human reading by trying to determine how two entities are connected to each other in a text corpus.

To conduct the Experiments we extracted the candidate relations using a prototype application following the sequence of steps outlined below.

1. Run the prototype application *RelationExtraction.mdb*, the main menu appears as shown.

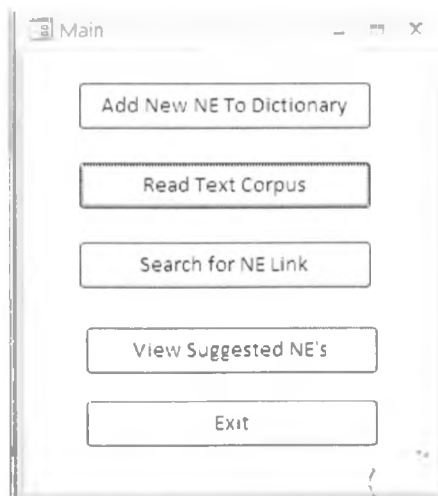


Figure 8: Prototype System Main Menu

2. Select “*Read Text Corpus*” from the Main Menu. The window shown in figure 9 appears. Specify “*Directory to Read text From*” and “*Directory to Output to*”. The “*Directory to Read text From*” specifies the location of the Test Corpus, whereas the “*Directory to Output to*” specifies the location that the candidate relations extracted by the Relational Extractor will be stored.

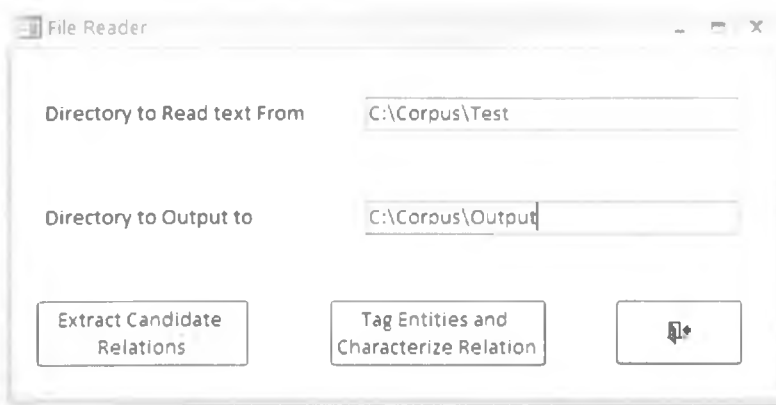


Figure 9: Prototype System File Reader

3. Click on “*Extract Candidate Relations*” button, this runs ReVerb. To confirm that the candidate relations have been extracted open the folder specified in the “*Directory to Output to*”.
4. Once it was confirmed that all the files that were specified in the “*Directory to read text from*” had been processed and corresponding output files created in the “*Directory to Output to*”, we then conducted the following experiments.

4.4.1 Experiment 1

Input

Ten (10) files extracted from the Test corpus using ReVerb i.e. files contained in the path specified by “*Directory to Output to*”.

Parameters

Named Entity Dictionary containing zero (0) records.

Procedure

Click on the “*Tag Entities and Characterize Relation*” button, this action runs the *TagAndCharacterize.py* program.

Output

- I. **Detected Named Entities:-** No Entities were added to the Inverted Index table
- II. **Extracted Relations:** - No Entity Relations were extracted
- III. **Suggested Named Entities:** - Eighty one (81) Named Entity suggestions were entered into the suggestedNE table for consideration for the user to insert them and their types in the Named Entity table.

Table 14: Experiment 1 Output- Named Entity Suggestions

Entity Name	Entity Name	Entity Name
Arap	Ccu	Fortnight
Hague	Unless	Marsabit
Bosek	william ruto	marsabit ruto kipchumba
Kigen	Kenyatta	Kipchumba
kioko kilukumi	Munyori	Poarchers
co-accused	Buku	Meru
eldoret William	Odm	isaih nakoru
Ruto	Odinga	g3
kuniko ozaki	Mudavadi	picha lokitela
christine wyngaert	Eldoret	uhuru kenyatta kanu
Ogetto	odm jakoyo midiwo	day-long
francis muthaura	Midiwo	kanu justin muturi
2pm	kichwa tembo	chama cha mwananchi
time-lines	Ololoolo	Kenda
urp william ruto	ole naiguran	Raila
raila odinga	ak47 g3	uhuru eldoret William
langat magerer	Naiguran	Kibaki
Bett	Lolgorian	muthaura karim
Julius	aitong narok	Fatou
Kanu	Sitoka	Tsavo
Pnu	olosentu laila	ak-47
Udf	Trans	tsavo korir
Belgut	Brian	Korir
Urp	24-hour	Sagalla
Orwa	Narok	taita ocpd nathaniel aseneka
chama cha uzalendo	Personel	
uhuru kenyatta	Whose	
ngari ccu	Heritag	

Analysis of Experiment 1 Results

The prototype was not able to extract any Entity Relations, since it had no knowledge of what constitutes an Entity arising from the fact that the Named Entity Dictionary had no records. However, the Prototype application was able to make suggestions of names or words that could be

added to the Named Entity Dictionary (see Table 14) to enable the prototype in future runs be able to extract entity relations appropriately.

4.4.2 Experiment 2

Input

Ten (10) files extracted from the Test corpus using ReVerb i.e. files contained in the path specified by “*Directory to Output to*”.

Parameters

Named Entity Dictionary containing seventy three (73) records identified by the experimenter from the suggested named Entity table as suitable named entities as shown in Table 15.

Table 15: Experiment 2 Parameters- Named Entities and their Entity Type

Entity Name	Entity Type
Tsavo	LOC
Marsabit	LOC
Meru	LOC
Ololoolo	LOC
kichwa tembo	LOC
Eldoret	LOC
Lolgorian	LOC
Belgut	LOC
Sagalla	LOC
Hague	LOC
Laila	LOC
Olosetu	LOC
Sitoka	LOC
Aitong	LOC
Narok	LOC
Taita	LOC
Kanu	ORG
chama cha mwananchi	ORG
Kenda	ORG
Udf	ORG
Ccu	ORG
Urp	ORG
odm	ORG
chama cha uzalendo	ORG
Pnu	ORG
Orwa	PER
Bett	PER
Julius	PER
Uhuru	PER
Ozaki	PER
Bosek	PER
Kijen	PER
Kioko	PER
Kilukumi	PER

Entity Name	Entity Type
Magerer	PER
Ogetto	PER
Francis	PER
Kenyatta	PER
Raila	PER
Odinga	PER
Langat	PER
Ruto	PER
Isaiah	PER
Brian	PER
Aseneka	PER
Nathaniel	PER
Korir	PER
Fatou	PER
Karim	PER
Kibaki	PER
Muturi	PER
Justin	PER
Lokitela	PER
Mudavadi	PER
Nakoru	PER
Ngari	PER
Kipchumba	PER
Naiguran	PER
Ole	PER
Midiwo	PER
Jakoyo	PER
Arap	PER
Wyngaert	PER
Christine	PER
Buku	PER
Munyori	PER
Picha	PER
g3	WEA

Entity Name	Entity Type
William	PER
Muthaura	PER
Kuniko	PER

Entity Name	Entity Type
ak47	WEA
ak-47	WEA

Procedure

1. From the Main Menu select “*Add New NE To Dictionary*”. The window shown below appears.

Figure 10: Prototype System -Add New Named Entity Window

2. Enter the Entities and types specified in Tale 15. Close the window once all entities have been keyed in.
3. Select “*Read Text Corpus*” from the Main Menu. Specify the location that the candidate extractions from the Relational Extractor are stored. In this case C:\Corpus\Output.
4. Click on the “*Tag Entities and Characterize Relation*” button, to run the *TagAndCharacterize.py* program.

Output

1. **Detected Named Entities:** -Sixty (60) Entities were detected and inserted into the Inverted Index table as shown.

Table 16: Experiment 2 Output - Inverted Index (Detected Entities)

Entity Name	Entity Type	Entity Name	Entity Type
Arap	PER	Ololoolo	LOC
Hague	LOC	Ole Naiguran	PER
Bosek	PER	Ak47	WEA
Kigen	PER	Naiguran	PER
Kioko Kilukumi	PER	Lolgorian	LOC
Eldoret	LOC	Aitong	LOC
Ruto	PER	Sitoka	LOC
Kuniko Ozaki	PER	Oloentu	LOC
Christine	PER	Brian	PER
Ogetto	PER	Marsabit	LOC
Francis Muthaura	PER	Ruto Kipchumba	PER
Urp	ORG	Kipchumba	PER
William Ruto	PER	Meru	LOC
Langat Magerer	PER	Nakoru	PER
Bett	PER	G3	WEA
Julius	PER	Picha Lokitela	PER
Kanu	ORG	Chama Cha Mwananchi	ORG
Pnu	ORG	Kenda	ORG
Belgut	LOC	Raila	PER
Chama Cha Uzalendo	ORG	Uhuru	PER
Uhuru Kenyatta	PER	William	PER
Ngari	PER	Wyngaert	PER
Ccu	ORG	Muthaura	PER
Raila Odinga	PER	Tsavo	LOC
Kenyatta	PER	Ak-47	WEA
Munyori	PER	Korir	PER
Buku	PER	Sagalla	LOC
Odm	ORG	Taita	LOC
Odinga	PER	Nathaniel Aseneka	PER
Jakoyo Midiwo	PER		
Midiwo	PER		

II. **Extracted Relations:** - Thirty one (31) Entity Relations were extracted as shown in the table.

Table 17: Experiment 2 Output - Extracted Entity Relations

Primary Entity Name	Relation	Secondary Entity Name
Arap	will travel to	Hague
Kigen	will travel with	Kioko Kilukumi
Kioko Kilukumi	Represent	Eldoret
Ruto	have requested	Kuniko Ozaki
Ogetto	who is	Francis Muthaura
Langat Magerer	has been a strong supporter of the pm	Ruto
Langat Magerer	Said	Ruto
Belgut	Keter	Urp
Chama Cha Uzalendo	would not support	Uhuru Kenyatta
Ccu	also said	Kanu
Ccu	was not party to	Kanu
William Ruto	dismissed allegations of a plot	Raila Odinga
Buku	was referring to	Odm
Odm	which has threatened to	Odinga
Ruto	Said	Kenyatta
Ruto	Described	Midiwo
Ruto	spoke in	Eldoret
Eldoret	presided over	Eldoret
Ole Naiguran	And	Ak47
Naiguran	Said	Lolgorian
Naiguran	were from	Lolgorian
Lolgorian	Hired	Aitong
Lolgorian	were headed for	Aitong
Naiguran	Said	Sitoka
Naiguran	are part of a dreaded poaching gang	Sitoka
Kipchumba	Said	Marsabit
Kipchumba	collude with	Marsabit
Kipchumba	Invade	Marsabit
Meru	have killed	G3
Meru	Recovered	G3
Wyngaert	Eboe-Osuji	Muthaura

III. **Suggested Named Entities:-** Fourteen (14) Named Entity suggestions were inserted into the suggestedNE table for consideration for the user to insert them and their types in the Named Entity table.

Table 18: Experiment 2 Output- Named Entity Suggestions

Entity Name	Entity Name
co-accused	whose
2pm	heritag
time-lines	fortinight
Unless	poarchers
Trans	day-long
24-hour	kichwa tembo
Personel	chama cha uzalendo

Analysis of Experiment 2 Results

The named entity suggestions from Experiment 1, that were included as parameters of Experiment 2 were unable to cover all entity name mentions. This is evidenced by the fact that the Inverted Index table contained only few full names for persons. We noted that English names such as Joshua, Philemon, Joel, Charles etc and vernacular names such as Sang, were not part of the suggested named entities in Experiment 1 (see Table 14) because the WordNet lexical dictionary used in the prototype application positively identified them as English words and therefore not candidates for inclusion into the Named Entity Dictionary. We therefore argue that the limited size of the named entity dictionary affected the performance of the prototype.

The named entity suggestions from Experiment 2 show that there are English words that the WordNet dictionary does not recognize and therefore suggests them as candidates for inclusion into the named entity dictionary (see Table 18). We therefore say that the size of the English dictionary that is used affects the quality of extractions, depending on words it contains or omits.

Further, misspelled words such as *poarchers*, *fortinight* were identified as candidate suggestions for inclusion into the named entity dictionary. This means that the underlying POS taggers identified the words as proper noun phrases, which is misleading and may result in incorrect relation extractions by REVERB and consequently incorrect characterization by the Prototype. We therefore argue that the quality of the written text affects the performance of the prototype application.

4.4.3 Experiment 3

Input

Ten (10) files extracted from the Test corpus using REVERB i.e. files contained in the path specified by “*Directory to Output to*”.

Parameters

A Named Entity Dictionary containing 50,000+ names sampled from a Person names register of Kenyan vernacular names and Common English names, Kenyan counties, Towns and City names and the seventy three (73) entity names identified in Experiment 2.

Procedure

1. A list of names was imported into the *namedEntity* table in Microsoft Access database with their Entity types specified.
2. Select “*Read Text Corpus*” from the Main Menu. Specify the location that the candidate extractions from the Relational Extractor are stored. In this case C:\Corpus\Output.
3. Click on the “*Tag Entities and Characterize Relation*” button, to run the *TagAndCharacterize.py* program.

Output

1. **Detected Named Entities:** -Ninety three (93) Entities were detected and inserted into the Inverted Index table as shown in Table 19.

Table 19: Experiment 3 Output - Inverted Index (Detected Entities)

Entity Type	Entity Name	Entity Type	Entity Name	Entity Type	Entity Name
PER	Joshua Arap Sang	PER	Buku	PER	Muthaura
LOC	Hague	ORG	Odm	PER	Khan
PER	Sang	PER	Odinga	PER	June
PER	Philemon	PER	Chief	LOC	Tsavo
PER	Bosek	PER	Jakoyo Midiwo	PER	One
PER	Are	PER	Sam	WEA	Ak-47
PER	Kigen	PER	Midiwo	PER	Wilson Korir
PER	Kioko Kilukumi	PER	Kichwa Tembo	PER	Korir
LOC	Eldoret	PER	Masai	LOC	Sagalla
PER	Ruto	LOC	Mara	PER	Same
PER	Kuniko Ozaki	LOC	Ololoolo	PER	Major
PER	Christine	PER	Gate	LOC	Taita
PER	Ken Ogetto	PER	Night	PER	Nathaniel Aseneka
PER	Francis Muthaura	PER	Scout		
PER	Time	PER	Joshua Ole Naiguran		
LOC	Rift Valley	WEA	Ak47		
ORG	Urp	PER	Naiguran		
PER	William Ruto	LOC	Lolgorian		
PER	Raila Odinga	LOC	Aitong		
PER	Langat Magerer	LOC	Sitoka		
PER	Franklin Bett	LOC	Oloentu		
PER	Julius	PER	Laila		
PER	Joyce	PER	Brian		
ORG	Kanu	LOC	Marsabit		
ORG	Pnu	PER	Ruto Kipchumba		
PER	More	PER	Kipchumba		
LOC	Nairobi	PER	Kenya		
LOC	Belgut	PER	Meru		
PER	Charles	PER	Isaih Nakoru		
PER	Said	WEA	G3		
PER	Some	PER	Picha Lokitela		
PER	George	ORG	Kws		
ORG	Chama Cha Uzalendo		Chama Cha		
PER	Uhuru Kenyatta	ORG	Mwananchi		
PER	Johnson Ngari	ORG	Kenda		
ORG	Ccu	PER	Same Time		
ORG	Icc	PER	Raila		
PER	Kenyatta	PER	Uhuru		
PER	Munyori	PER	William		
PER	Were	PER	December		
		PER	Wyngaert		

II. **Extracted Relations:** - Fifty one (51) Entity Relations were extracted as shown in the table.

Table 20: Experiment 3 Output - Extracted Entity Relations

Primary Entity Name	Relation	Secondary Entity Name
Joshua Arap Sang	will travel to	Hague
Bosek	Katwa Kigen have already	Are
Kigen	will travel with	Kioko Kilukumi
Kioko Kilukumi	represent	Sang
Ruto	have requested	Kuniko Ozaki
Ken Ogetto	who is	Francis Muthaura
Rift Valley	allied to	Urp
Langat Magerer	has been a strong supporter of the pm	Ruto
Langat Magerer	said	Ruto
Ruto	were on	More
Charles	Keter	Said
Chama Cha Uzalendo	would not support	Uhuru Kenyatta
Ccu	also said	Kanu
Ccu	was not party to	Kanu
Uhuru Kenyatta	is facing serious charges of crimes	Icc
Icc	suspects	Uhuru Kenyatta
William Ruto	dismissed allegations of a plot	Raila Odinga
Munyori	,	Were
Buku	was referring to	Odm
Odm	which has threatened to	Odinga
Ruto	said	Kenyatta
Ruto	described	Midiwo
Ruto	spoke in	Eldoret
Eldoret	presided over	Eldoret
Gate	were on	Night
Night	acted on	Scout
Joshua Ole Naiguran	and	Ak47
Naiguran	said	Lolgorian
Naiguran	were from	Lolgorian
Lolgorian	hired	Aitong
Lolgorian	were headed for	Aitong
Naiguran	said	Sitoka
Naiguran	are part of a dreaded poaching gang	Sitoka
Brian	,	Said
More	have been killed in	Marsabit
Kipchumba	said	Marsabit
Kipchumba	collude with	Marsabit
Kipchumba	invade	Marsabit

Primary Entity Name	Relation	Secondary Entity Name
Kenya	support	Marsabit
Kenya	comprises	Marsabit
Meru	have killed	G3
Meru	recovered	G3
Kws		Said
Chama Cha Uzalendo	Kenya and new revival generation	Are
Francis Muthaura	have lost	Icc
Wyngaert	Eboe-Osuji	Muthaura
Khan	had during	June
Kws	were killed at	Sagalla
Korir	poachers could	Same
Same	who had	Kws
Night	mounted	Said

III. **Suggested Named Entities:** - Fourteen (14) Named Entity suggestions were inserted into the suggestedNE table for consideration for the user to insert them and their types in the Named Entity table.

Table 21: Experiment 3 Output- Named Entity Suggestions

Entity Name	Entity Name
co-accused	whose
2pm	heritag
time-lines	fortinight
Unless	poarchers
Trans	day-long
24-hour	kichwa tembo
Personel	chama cha uzalendo

Analysis of Experiment 3 Results

The increased size of the entity name dictionary resulted in an increase in the number of named entities detected i.e. from sixty (60) to ninety three (93). English names such as Joshua, Charles and Philemon were detected. Additionally vernacular names such as Sang were also correctly detected. However, the following set of names was identified to have been incorrectly detected, they are: *Are, Were, One, Time, More, Said, Were, Gate, Night, Scout, Kenya, Meru, December, June, Major, Same, Chief and Masai.*

The incorrect detection of entities saw number of incorrect entity relations characterized increase by 18 as shown in the table below.

Table 22: Experiment 3- Incorrect Entity Relation Characterization

Primary Entity Name	Relation	Secondary Entity Name
Bosek	Katwa Kigen have already	Are
Ruto	were on	More
Charles	Keter	Said
Munyori	,	Were
Gate	were on	Night
Night	acted on	Scout
Brian	,	Said
More	have been killed in	Marsabit
Kenya	support	Marsabit
Kenya	comprises	Marsabit
Meru	have killed	G3
Meru	recovered	G3
Kws	,	Said
Chama Cha Uzalendo	Kenya and new revival generation	Are
Khan	had during	June
Korir	poachers could	Same
Same	who had	Kws
Night	mounted	Said

The increase in incorrect characterization of entity relations covering the domains of politics, poaching and accidents brought forth the idea of trimming the name entity dictionary to suit a particular subject of interest and hence the possibility of having different named entity dictionaries each covering a specific domain.

We put to test this possibility in Experiment 4 by narrowing the domain covered by the characterizer to politics.

4.4.4 Experiment 4

Input

Four files extracted from the model Test corpus's Ten (10) news stories using REVERB i.e. Four (4) candidate relation files. The four files cover Kenyan politics revolving around the International Criminal Court proceedings.

Parameters

Named Entity Dictionary containing 50,000+ names sampled from a Person names register, County names, Town names, City names and including seventy three (73) entity names identified in Experiment 2.

Procedure

1. From the Main Menu select "Add New NE To Dictionary". Search and delete the names that were identified to have been incorrectly detected in Experiment 3. The names are: *Are, Were, One, Time, More, Said, Gate, Night, Scout, Kenya, Meru, Same, December, June, Major, Some, Chief and Masai*.
2. Select "Read Text Corpus" from the Main Menu. Specify the location that the candidate extractions from the Relational Extractor are stored. In this case C:\Corpus\Output.
3. Click on the "Tag Entities and Characterize Relation" button, to run the *TagAndCharacterize.py* program.

Output

For this experiment we outline of each of the four (4) news stories as they were published (see Tables 23,26,29,32), we then tabulate annotator's A & B annotation for each story (see Tables 24,27,30,33) and calculate the IAA for each, we then tabulated the annotators and prototype extractions (see Tables 25,28,31,34) and calculate the Precision, Error rate and Recall for each.

The annotations were done by double underlining the Named Entity and single underlining the relation between them. Below are the news stories and their corresponding annotations.

Finally, we present a summary tabulation of the actual extractions by the prototype application in Tables 35 and 36, representing the detected named entities and the extracted entity relations respectively.

Table 23: Nation newspaper news story of 6th June 2012

File Name: Nation-06-6-2012.txt

Author: Oliver Mathenge

Radio presenter Joshua arap Sang and his three lawyers will travel to The Hague on Saturday for a status conference ahead of his trial at the International Criminal Court. According to Mr Sang, lawyers Philemon Koech, Joel Bosek and Katwa Kigen have already applied for their visas and are awaiting clearance from the Dutch Embassy in Nairobi.

Mr Kigen will travel with Mr Kioko Kilukumi with whom they represent Mr Sang's co-accused, Eldoret North MP William Ruto. The lawyers did not say whether Mr Ruto would be travelling but sources close to the politician have disclosed that he would not be going.

The prosecution, the defence, the ICC registry and the victims' lawyers in the case against Mr Ruto and Mr Sang will attend the status conference on Monday where the trial date for the two will be set. Mr Sang and Mr Ruto have requested Trial Chamber judges Kuniko Ozaki, Christine Van den Wyngaert and Chile Eboe-Osuji to have the trial date set after the next General Election. Mr Ken Ogetto, who is in former Head of Civil Service Francis Muthaura's defence team, will also be travelling on Saturday for the status conference in the second case which will be held on Tuesday.

Lawyers for Mr Muthaura's co-accused, Deputy Prime Minister Uhuru Kenyatta, were unavailable for comment but are also expected to travel over the weekend. Sources have said that the two will not travel to The Hague. Both meetings will begin at 2pm Kenyan time but the suspects are not required to be in court in person.

On average, it has taken between six and eight months for previous cases to start after the status conferences. The judges have asked parties to the case to make any submissions regarding the agenda of the status conferences. "If the parties, the legal representatives of victims and the registry are currently aware of any other issue that is required to be resolved before the commencement of the trial, they should bring it to the attention of the Chamber promptly," the Trial Chamber judges said.

Apart from setting the date of trials, the conference will also set the time-lines and format of disclosing evidence including witnesses who will require protection. The prosecution has indicated it will require a year to present its evidence in each of the cases.

Table 24: Annotator A & B annotation of Nation newspaper story of 6th June 2012

File Name: Nation-06-6-2012.txt			
Sentence	Annotator		Agree
	A	B	
Radio presenter <u>Joshua arap Sang</u> and his three lawyers will travel to <u>The Hague</u> on Saturday for a status conference ahead of his trial at the International Criminal Court.	Yes	Yes	Yes
According to Mr Sang, lawyers Philemon Koech, Joel Bosek and <u>Katwa Kigen</u> have already <u>applied for their visas</u> and are awaiting clearance from the <u>Dutch Embassy</u> in Nairobi.	Yes	Yes	Yes
<u>Mr Kigen</u> will travel with <u>Mr Kioko Kilukumi</u> with whom they represent Mr Sang's co-accused, Eldoret North MP William Ruto.	Yes	Yes	Yes
Mr Kigen will travel with Mr Kioko Kilukumi with whom they represent <u>Mr Sang's co-accused</u> , Eldoret North MP <u>William Ruto</u> .	Yes	No	No
Mr Sang and <u>Mr Ruto</u> have <u>requested</u> Trial Chamber judges <u>Kuniko Ozaki</u> , Christine Van den Wyngaert and Chile Eboe-Osuji to have the trial date set after the next General Election.	Yes	Yes	Yes
<u>Mr Ken Ogetto</u> , who is in former Head of Civil Service <u>Francis Muthaura's</u> defence team, will also be travelling on Saturday for the status conference in the second case which will be held on Tuesday.	Yes	Yes	Yes
Lawyers for <u>Mr Muthaura's co-accused</u> , Deputy Prime Minister <u>Uhuru Kenyatta</u> , were unavailable for comment but are also expected to travel over the weekend.	Yes	Yes	Yes

To calculate IAA for the above we use the chance-corrected agreement measure R, whereby A_o is the observed (or “percentage”) agreement and A_e is expected agreement by chance, which is 50% since there are two annotators.

$$R = \frac{A_o - A_e}{1 - A_e}$$

In the above case A_o = 6/7 x 100 = 85.7

$$R = \frac{85.7 - 50}{100 - 50}$$

$$R = 0.714$$

Table 25: Annotator and Prototype Extractions of the Nation news story of 6th June 2012

File Name: Nation-06-6-2012.txt					
Sentence	Annotator		Ann. Agree	Prototype Extracted	Correct Extraction
	A	B			
Radio presenter <u>Joshua arap Sang</u> and his three lawyers <u>will travel to The Hague</u> on Saturday for a status conference ahead of his trial at the International Criminal Court.	Yes	Yes	Yes	Yes	Yes
According to Mr Sang, lawyers <u>Philemon Koech</u> , <u>Joel Bosek</u> and <u>Katwa Kigen</u> have already <u>applied for their visas</u> and are awaiting clearance from the <u>Dutch Embassy</u> in Nairobi.	Yes	Yes	Yes	No	No
<u>Mr Kigen</u> will travel with <u>Mr Kioko Kilukumi</u> with whom they represent Mr Sang's co-accused, Eldoret North MP <u>William Ruto</u> .	Yes	Yes	Yes	Yes	Yes
Mr Kigen will travel with <u>Mr Kioko Kilukumi</u> with whom they <u>represent Mr Sang's</u> co-accused, Eldoret North MP <u>William Ruto</u> .	No	No	No	Yes	No
Mr Sang and <u>Mr Ruto</u> <u>have requested</u> Trial Chamber judges <u>Kuniko Ozaki</u> , <u>Christine Van den Wyngaert</u> and <u>Chile Eboe-Osuji</u> to have the trial date set after the next General Election.	Yes	Yes	Yes	Yes	Yes
<u>Mr Ken Ogetto</u> , <u>who is</u> in former Head of Civil Service <u>Francis Muthaura's</u> defence team, will also be travelling on Saturday for the status conference in the second case which will be held on Tuesday.	Yes	Yes	Yes	Yes	Yes
Lawyers for <u>Mr Muthaura's</u> <u>co-accused</u> , Deputy Prime Minister <u>Uhuru Kenyatta</u> , were unavailable for comment but are also expected to travel over the weekend.	Yes	Yes	Yes	No	No

Total relations extracted by the human annotators (T_H) = 6. Total relations extracted by Prototype system (T_P) = 5. Total number of correct relations extracted (T_C) = 4.

$$Precision = \frac{T_c}{T_p} \times 100 = \frac{4}{5} \times 100 = 80$$

$$Error\ rate = \frac{T_p - T_c}{T_p} \times 100 = \frac{5 - 4}{5} \times 100 = 20$$

$$Recall = \frac{T_c}{T_H} \times 100 = \frac{4}{6} \times 100 = 66.67$$

Table 26: Nation newspaper news story of 17th March 2012

File Name: Nation-17-3-2012.txt	Author:
<p>Chama Cha Uzalendo yesterday said that it would not support Mr Uhuru Kenyatta's presidential bid. In a statement signed by Spokesman Johnson Ngari CCU also said it was not party to an agreement reportedly signed by 15 parties to form an alliance with kanu. "CCU is aware that Uhuru Kenyatta is facing serious charges of crimes against humanity at the ICC and it will therefore, be naive for the party to enter into an alliance with such a person unless fully cleared".</p>	

Table 27: Annotator A & B annotation of Nation newspaper story of 17th March 2012

File Name: Nation-17-3-2012.txt			
Sentence	Annotator		Agree
	A	B	
<u>Chama Cha Uzalendo</u> yesterday said that it <u>would not support Mr Uhuru Kenyatta's</u> presidential bid.	Yes	Yes	Yes
In a statement signed by Spokesman Johnson Ngari <u>CCU</u> also said it was <u>not party to an agreement</u> reportedly signed by 15 parties to form an alliance with <u>kanu</u> .	Yes	Yes	Yes
"CCU is aware that <u>Uhuru Kenyatta</u> is <u>facing serious charges of crimes</u> against humanity at the <u>ICC</u> and it will therefore, be naive for the party to enter into an alliance with such a person unless fully cleared".	Yes	Yes	Yes

In the above case $A_o = 3/3 \times 100 = 100$

$$R = \frac{100 - 50}{100 - 50} = 1$$

Table 28: Annotator & Prototype Extractions of the Nation news story of 17th Mar 2012

File Name: Nation-17-3-2012.txt					
Sentence	Annotator		Ann. Agree	Prototype Extracted	Correct Extraction
	A	B			
<u>Chama Cha Uzalendo</u> yesterday said that it would not support <u>Mr Uhuru Kenyatta's</u> presidential bid.	Yes	Yes	Yes	Yes	Yes
In a statement signed by Spokesman Johnson Ngari <u>CCU</u> also said it was <u>not party to an agreement</u> reportedly signed by 15 parties to form an alliance with <u>kanu</u> .	Yes	Yes	Yes	Yes	Yes
In a statement signed by Spokesman Johnson Ngari <u>CCU</u> also said it was not party to an agreement reportedly signed by 15 parties to form an alliance with <u>kanu</u> .	No	No	No	Yes	No
"CCU is aware that <u>Uhuru Kenyatta</u> is <u>facing serious charges of crimes</u> against humanity at the <u>ICC</u> and it will therefore, be naive for the party to enter into an alliance with such a person unless fully cleared".	Yes	Yes	Yes	Yes	Yes

$$Precision = \frac{T_c}{T_p} \times 100 = \frac{3}{4} \times 100 = 75$$

$$Error\ rate = \frac{T_p - T_c}{T_p} \times 100 = \frac{4 - 3}{4} \times 100 = 25$$

$$Recall = \frac{T_c}{T_H} \times 100 = \frac{3}{3} \times 100 = 100$$

Table 29: Nation newspaper news story of 18th April 2012

File Name: Nation-18-04-2012.txt	Author: Fred Mukinda and Samuel Koech
---	--

ICC suspects Uhuru Kenyatta and William Ruto on Tuesday dismissed allegations of a plot to assassinate Prime Minister Raila Odinga. Mr Kenyatta's spokesman, Mr Munyori Buku, dismissed the claims, saying that they were part of a "scheme to camouflage the biggest failure in that party (Orange Democratic Movement) - dictatorship and resistance to change". "This issue of dropping names of people holding senior positions recklessly, instead of working for Kenyans, is a scheme that borders on devilish acts," said Mr Buku.

He added: "They should spend more energy resolving their party disputes. There is room for competitive politics without making defamatory statements." Mr Buku was referring to nomination of ODM's presidential candidate, which has threatened to split the party, with some members insisting on unchallenged nomination of Mr Odinga as party leader while those allied to his deputy, Mr Musalia Mudavadi, proposing a competitive process to pick a contender.

Mr Ruto, the Eldoret North MP, said separately that it was unfortunate that people can make casual statements on sensitive matters pertaining to security. He urged Kenyans to treat ODM chief whip Jakoyo Midiwo's claims that there was a plot to kill Mr Odinga with the "contempt they deserve". "It is so unfortunate that people can make alarming, unsubstantiated statements that have no basis and are bound to cause unnecessary friction, especially at this time when the country is gearing for the general election," said Mr Ruto.

Mr Ruto said Deputy Prime Minister Kenyatta, Foreign Affairs minister Sam Ongerir and himself had recorded statements with the police and urged security agencies to move with haste in investigating the motive behind the claims. Mr Ruto described Mr Midiwo's claims as a "useless story", adding that it is only a mad man who can think of hatching a plot to assassinate the Prime Minister. Mr Ruto spoke in Eldoret yesterday when he presided over the Eldoret West District education and prize giving day. "If the PM can be at risk with the heavy security detail attached to him, then what about the ordinary Kenyan citizen?" asked Mr Ruto.

Table 30: Annotator A & B annotation of Nation newspaper story of 18th April 2012

File Name: Nation-18-04-2012.txt			
Sentence	Annotator		Agree
	A	B	
ICC suspects <u>Uhuru Kenyatta</u> and <u>William Ruto</u> on Tuesday dismissed allegations of a plot to assassinate Prime Minister Raila Odinga.	Yes	Yes	Yes
ICC suspects <u>Uhuru Kenyatta</u> and <u>William Ruto</u> on Tuesday <u>dismissed allegations of a plot to assassinate Prime Minister Raila Odinga.</u>	Yes	Yes	Yes
<u>Mr Kenyatta's spokesman, Mr Muniyori Buku,</u> dismissed the claims, saying that they were part of a "scheme to camouflage the biggest failure in that party (Orange Democratic Movement) - dictatorship and resistance to change".	Yes	Yes	Yes
<u>Mr Buku was referring to nomination of ODM's presidential candidate,</u> which has threatened to split the party, with some members insisting on unchallenged nomination of Mr Odinga as party leader while those allied to his deputy, Mr Musalia Mudavadi, proposing a competitive process to pick a contender.	Yes	Yes	Yes
<u>Mr Ruto, the Eldoret North MP,</u> said separately that it was unfortunate that people can make casual statements on sensitive matters pertaining to security.	Yes	No	No
He urged Kenyans to treat <u>ODM chief whip Jakoyo Midiwo's</u> claims that there was a plot to kill Mr Odinga with the "contempt they deserve".	Yes	Yes	Yes
<u>Mr Ruto described Mr Midiwo's</u> claims as a "useless story", adding that it is only a mad man who can think of hatching a plot to assassinate the Prime Minister.	Yes	Yes	Yes
<u>Mr Ruto spoke in Eldoret</u> yesterday when he presided over the Eldoret West District education and prize giving day.	Yes	Yes	Yes

In the above case $A_o = 7/8 \times 100 = 87.5$

$$R = \frac{87.5 - 50}{100 - 50} = 0.75$$

Table 31: Annotator & Prototype Extractions of the Nation news story of 18th April 2012

File Name: Nation-18-04-2012.txt

Sentence	Annotator		Ann. Agree	Prototype Extracted	Correct Extraction
	A	B			
ICC suspects <u>Uhuru Kenyatta</u> and <u>William Ruto</u> on Tuesday dismissed allegations of a plot to assassinate Prime Minister Raila Odinga.	Yes	Yes	Yes	Yes	Yes
ICC suspects <u>Uhuru Kenyatta</u> and <u>William Ruto</u> on Tuesday dismissed allegations of a plot to assassinate Prime Minister Raila Odinga.	Yes	Yes	Yes	Yes	Yes
Mr Kenyatta's spokesman, Mr Munyori Buku, dismissed the claims, saying that they were part of a "scheme to camouflage the biggest failure in that party (Orange Democratic Movement) - dictatorship and resistance to change".	Yes	Yes	Yes	No	No
Mr Buku was referring to nomination of ODM's presidential candidate, which has threatened to split the party, with some members insisting on unchallenged nomination of Mr Odinga as party leader while those allied to his deputy, Mr Musalia Mudavadi, proposing a competitive process to pick a contender.	Yes	Yes	Yes	Yes	Yes
Mr Buku was referring to nomination of ODM's presidential candidate, which has threatened to split the party, with some members insisting on unchallenged nomination of Mr Odinga as party leader while those allied to his deputy, Mr Musalia Mudavadi, proposing a competitive process to pick a contender.	No	No	No	Yes	No
Mr Ruto, the Eldoret North MP, said separately that it was unfortunate that people can make casual statements on sensitive matters pertaining to security.	Yes	No	No	No	No
He urged Kenyans to treat ODM chief whip <u>Jakoyo Midiwo's</u> claims that there was a plot to kill Mr Odinga with the "contempt they deserve".	Yes	Yes	Yes	No	No
Mr Ruto said Deputy Prime Minister <u>Kenyatta</u> , Foreign Affairs minister Sam Ongeru and himself had recorded statements with the police and urged security agencies to move with haste in investigating the motive behind the claims.	No	No	No	Yes	No
Mr Ruto described Mr Midiwo's claims as a "useless story", adding that it is only a mad man who can think of hatching a plot to assassinate the Prime Minister.	Yes	Yes	Yes	Yes	Yes
Mr Ruto spoke in Eldoret yesterday when he presided over the Eldoret West District education and prize giving day.	Yes	Yes	Yes	Yes	Yes
Mr Ruto spoke in Eldoret yesterday when he presided over the Eldoret West District education and prize giving day.	No	No	No	Yes	No

$$\text{Precision} = \frac{T_c}{T_p} \times 100 = \frac{5}{8} \times 100 = 62.5$$

$$\text{Error rate} = \frac{T_p - T_c}{T_p} \times 100 = \frac{8 - 5}{8} \times 100 = 37.5$$

$$\text{Recall} = \frac{T_c}{T_H} \times 100 = \frac{5}{7} \times 100 = 71.4$$

Table 32: An excerpt of The Star newspaper story of 23rd June 2012

File Name: Star-23-06-2012.txt	Author: Nzau Musau
<p>Deputy Prime Minister Uhuru Kenyatta and former head of civil service Francis Muthaura have lost a bid to gag the ICC prosecutor from contacting their witnesses. Trial judges Kuniko Ozaki (presiding judge), Christine Van den Wyngaert and Eboe-Osuji rejected the application initially sought by Muthaura's lawyer Karim Khan and later supported by Uhuru.</p> <p>The two are facing trial at the court over crimes against humanity. They are charged alongside Eldoret North MP William Ruto and radio presenter Joshua arap Sang. Khan had during the status conference held on June 12, applied to the court issue an interim order prohibiting the prosecution now led by Fatou Bensouda from contacting potential witnesses until the judges issue a ruling on the procedure of contacting witnesses.</p>	

Table 33: Annotator A&B annotation of the The Star news story excerpt of 23rd June 2012

File Name: Star-23-06-2012.txt			
Sentence	Annotator		Agree
	A	B	
Deputy Prime Minister Uhuru Kenyatta and former head of civil service <u>Francis Muthaura</u> have <u>lost a bid to gag</u> the <u>ICC</u> prosecutor from contacting their witnesses.	Yes	Yes	Yes
Trial judges Kuniko Ozaki (presiding judge), Christine Van den Wyngaert and <u>Eboe-Osuji</u> <u>rejected the application</u> initially sought by <u>Muthaura's</u> lawyer Karim Khan and later supported by Uhuru.	Yes	Yes	Yes
<u>Khan</u> had during the status conference held on June 12, <u>applied to the court</u> <u>issue an interim</u> order prohibiting the prosecution now led by <u>Fatou Bensouda</u> from contacting potential witnesses until the judges issue a ruling on the procedure of contacting witnesses.	Yes	Yes	Yes

In the above case $A_o = 3/3 \times 100 = 100$

$$R = \frac{100 - 50}{100 - 50} = 1$$

Table 34: Annotator & Prototype Extractions of The Star news story of 23rd June 2012

File Name: Star-23-06-2012.txt					
Sentence	Annotator		Ann. Agree	Prototype Extracted	Correct Extraction
	A	B			
Deputy Prime Minister Uhuru Kenyatta and former head of civil service <u>Francis Muthaura</u> have <u>lost a bid to gag</u> the <u>ICC</u> prosecutor from contacting their witnesses.	Yes	Yes	Yes	Yes	Yes
Trial judges Kuniko Ozaki (presiding judge), Christine Van den Wyngaert and <u>Eboe-Osuji</u> <u>rejected the application</u> initially sought by <u>Muthaura's</u> lawyer Karim Khan and later supported by Uhuru.	Yes	Yes	Yes	No	No
Trial judges Kuniko Ozaki (presiding judge), Christine Van den <u>Wyngaert</u> and <u>Eboe-Osuji</u> rejected the application initially sought by <u>Muthaura's</u> lawyer Karim Khan and later supported by Uhuru.	No	No	No	Yes	No
<u>Khan</u> had during the status conference held on June 12, <u>applied to the court</u> issue an <u>interim order</u> prohibiting the prosecution now led by <u>Fatou Bensouda</u> from contacting potential witnesses until the judges issue a ruling on the procedure of contacting witnesses.	Yes	Yes	Yes	Yes	Yes

$$Precision = \frac{T_c}{T_p} \times 100 = \frac{2}{3} \times 100 = 66.7$$

$$Error\ rate = \frac{T_p - T_c}{T_p} \times 100 = \frac{3 - 2}{3} \times 100 = 33.3$$

$$Recall = \frac{T_c}{T_H} \times 100 = \frac{2}{3} \times 100 = 66.7$$

- I. **Detected Named Entities:** -Twenty nine (29) Entities were detected and inserted into the Inverted Index table as shown.

Table 35: Experiment 4 Output - Inverted Index (Detected Entities)

EntityType	EntityName	EntityType	EntityName
PER	Joshua Arap Sang	PER	Johnson Ngari
LOC	Hague	ORG	Ccu
PER	Sang	ORG	Kanu
PER	Philemon	ORG	Icc
PER	Bosek	PER	William Ruto
PER	Kigen	PER	Raila Odinga
PER	Kioko Kilukumi	PER	Kenyatta
LOC	Eldoret	PER	Munyori
PER	Ruto	PER	Buku
PER	Kuniko Ozaki	ORG	Odm
PER	Christine	PER	Odinga
PER	Ken Ogetto	PER	Jakoyo Midiwo
PER	Francis Muthaura	PER	Sam
ORG	Chama Cha Uzalendo	PER	Midiwo
PER	Uhuru Kenyatta		

- II. **Extracted Relations:** - twenty (20) Entity Relations were extracted as shown in the table.

Table 36: Experiment 4 Output - Extracted Entity Relations

Primary Entity	Relation	Secondary Entity	File Name
Joshua Arap Sang	will travel to	Hague	c:\corpus\test\Nation-06-6-2012.txt
Kioko Kilukumi	will travel with	Kioko Kilukumi	c:\corpus\test\Nation-06-6-2012.txt
Kioko Kilukumi	represent	Sang	c:\corpus\test\Nation-06-6-2012.txt
Ruto	have requested	Kuniko Ozaki	c:\corpus\test\Nation-06-6-2012.txt
Ken Ogetto	who is	Francis Muthaura	c:\corpus\test\Nation-06-6-2012.txt
Chama Cha Uzalendo	would not support	Uhuru Kenyatta	c:\corpus\test\Nation-17-03-2012.txt
Ccu	also said	Kanu	c:\corpus\test\Nation-17-03-2012.txt
Ccu	was not party to	Kanu	c:\corpus\test\Nation-17-03-2012.txt
Uhuru Kenyatta	is facing serious charges of crimes	Icc	c:\corpus\test\Nation-17-03-2012.txt
Icc	suspects	Uhuru Kenyatta	c:\corpus\test\Nation-18-04-2012.txt
William Ruto	dismissed allegations of a plot	Raila Odinga	c:\corpus\test\Nation-18-04-2012.txt
Buku	was referring to	Odm	c:\corpus\test\Nation-18-04-2012.txt
Odm	which has threatened to	Odinga	c:\corpus\test\Nation-18-04-2012.txt
Ruto	said	Kenyatta	c:\corpus\test\Nation-18-04-2012.txt
Ruto	described	Midiwo	c:\corpus\test\Nation-18-04-2012.txt
Eldoret	spoke in	Eldoret	c:\corpus\test\Nation-18-04-2012.txt
Francis Muthaura	presided over	Eldoret	c:\corpus\test\Nation-18-04-2012.txt
Francis Muthaura	have lost	Icc	c:\corpus\test\Star-23-06-2012.txt
Francis Muthaura	Eboe-Osuji	Muthaura	c:\corpus\test\Star-23-06-2012.txt
Francis Muthaura	had during	Fatou	c:\corpus\test\Star-23-06-2012.txt

Analysis of Experiment 4 Results

Below is a summary tabulation of the Total number of relations extracted by human annotators (T_H), Total number of relations extracted by the prototype (T_P) and the Total Correct relations extracted (T_C).

Table 37: Summary of Experiment 4 results

File Name	T_H (annotator)	T_P (prototype)	T_C (Correct Extraction)
c:\corpus\test\Nation-06-6-2012.txt	6	5	4
c:\corpus\test\Nation-17-03-2012.txt	3	4	3
c:\corpus\test\Nation-18-04-2012.txt	7	8	5
c:\corpus\test\Star-23-06-2012.txt	3	3	2
Total	19	20	14

$$Precision = \frac{T_C}{T_P} \times 100 = \frac{14}{20} \times 100 = 70$$

$$Recall = \frac{T_C}{T_H} \times 100 = \frac{14}{19} \times 100 = 73.7$$

We note that the narrowing down of the corpus to focus on a particular subject of interest and trimming named entity dictionary had an impact on the correct characterization of entity relational facts. However the prototype was unable to extract and/or characterize some relations. Below is a summary of entity relational facts that annotators deemed to be correct but which the prototype did not extract or characterize. This can be represented by the formula:

$$Omissions = 1 - Recall$$

Table 38: List of Omitted Extractions

No.	Primary Entity	Relation	Secondary Entity	File Name
1.	Katwa Kigen	applied for their visas	Dutch Embassy	c:\corpus\test\Nation-06-6-2012.txt
2.	Mr Muthaura's	co-accused	Uhuru Kenyatta	c:\corpus\test\Nation-06-6-2012.txt
3.	Mr Kenyatta's	Spokesman	Munyoru Buku	c:\corpus\test\ Nation-18-04-2012.txt
4.	ODM	chief whip	Jakoyo Midiwo	c:\corpus\test\ Nation-18-04-2012.txt
5.	Eboe-Osuji	rejected the application	Muthaura	c:\corpus\test\ Star-23-06-2012.txt

For the items No. 1 and 5 we note that despite the fact that the Relational Extractor component of the prototype system identifying the lexico-syntactic patterns in the sentences containing the relations, the Named Entity Detector (NED) component could not tag the entities since the names "Dutch Embassy" and "Eboe-Osuji" were not contained in the Named Entity Dictionary.

For the items No. 2, 3 and 4 we note that the Relational Extractor component of the prototype system was not able to identify them as entity relational facts since the OpenNL Part of Speech tagger tagged the words “co-accused” as a proper noun, “spokesman” as a noun, “chief whip” as two nouns. This made the sentence pattern fail to conform to lexico-syntactic pattern as outlined in Table 3. It with this in mind that we conclude items 2,3, and 4 form part of the slightly over 5% binary relations that are not described by any of eight lexico-syntactic patterns described in Table 3.

From the fore going we can argue that Recall consists of two components:

- i) Comprehensiveness of the name entity dictionary which is largely dependent on striking a balance between overlapping English words, person names and location names.
- ii) The frequency of outlier entity relations i.e. entity relations that do not conform to the lexico-syntactic patterns in Table 3, within a selected text corpus.

We therefore conclude that Recall can only be increased up to a certain limit by having a comprehensive dictionary.

Below is a summary of incorrect extractions by the prototype

Table 39: List of Errorneous extractions

Primary Entity	Relation	Secondary Entity	File Name
Ccu	also said	Kanu	c:\corpus\test\Nation-17-03-2012.txt
Eldoret	presided over	Eldoret	c:\corpus\test\Nation-18-04-2012.txt
Wyngaert	Eboe-Osuji	Muthaura	c:\corpus\test\Star-23-06-2012.txt

Precision is a function the characterizer’s ability to determine a valid form of a relation between two entities. The complexity of a sentence and the existence of multiple lexico-syntactic patterns between two entity names may adversely affect precision.

4.4.5 Experiment 5

Input


Two named Entities that a user wishes to find out how they are related to each other.

Parameters

For this Experiment we use the Detected named entities and the Entity Relational Facts extracted from four (4) documents in Experiment 4 and represented by Table 35 and Table 36 respectively.

Procedure

From the Main Menu shown in Figure 8, select “*Search for NE Link*”. The window shown below appears. Select the Primary Entity Name and the Secondary Entity Name and then click on Search.



The screenshot shows a window titled "Entity Relation Search". It contains two dropdown menus labeled "EntityName 1" and "EntityName 2". Below these is a "Search" button. Underneath the search button is a table with the following columns: "Primary Entity", "Relation", "Secondary Entity", and "File Name". The table is currently empty. At the bottom right of the window is a small icon button.

Figure 11: Prototype System- Entity Search Dialog

Output

Table 40: Experiment 5 Output - Entity Link Search

INPUT: 1	Entity Name 1	Kigen		
	Entity Name 2	Sang		
OUTPUT: 1	Primary Entity	Relation	Sec. Entity	File Name
	Kigen	will travel with	Kioko Kilukumi	c:\corpus\test\Nation-06-6-2012.txt
	Kioko Kilukumi	represent	Sang	c:\corpus\test\Nation-06-6-2012.txt
INPUT: 2	Entity Name 1	Chama Cha Uzalendo		
	Entity Name 2	Icc		
OUTPUT: 2	Primary Entity	Relation	Sec. Entity	File Name
	Chama Cha	would not	Uhuru	
	Uzalendo	support	Kenyatta	c:\corpus\test\Nation-17-03-2012.txt
	Uhuru Kenyatta	is facing serious charges of crimes	Icc	c:\corpus\test\Nation-17-03-2012.txt
INPUT: 3	Entity Name 1	Ccu		
	Entity Name 2	Uhuru Kenyatta		
OUTPUT: 3	No link Discovered			

Analysis of Experiment 5 Results

The prototype was able to establish a link between two named entities that were consistently referenced in a document, however the prototype was not able to establish a link between two named entities that refer to the same Person or Organization when they are not consistently expressed e.g the prototype could not establish a link between Joshua arap Sang and Sang, Raila Odinga and Odinga, William Ruto and Ruto, Uhuru Kenyatta and Kenyatta, Chama Cha Uzalendo and CCU though the stated pairs were all contained in the same document and refer to the same Entity. Figure 12 below gives a graph representation of extracted entities contained in Table 36 we note that there are no interconnecting vertices between named entities mentioned above. This failure affects the conclusiveness of a search of how two named entities are linked within a unstructured text corpus.

To address this problem a mechanism for the prototype system to generate relations linking inconsistent entity name mentions can be adopted. Figure 13 gives a graph representation of extracted named entities contained in Table 36 with possible system generated vertices included.

Figure 12: Graph Representation of Extracted Entity Relations contained in Table 36

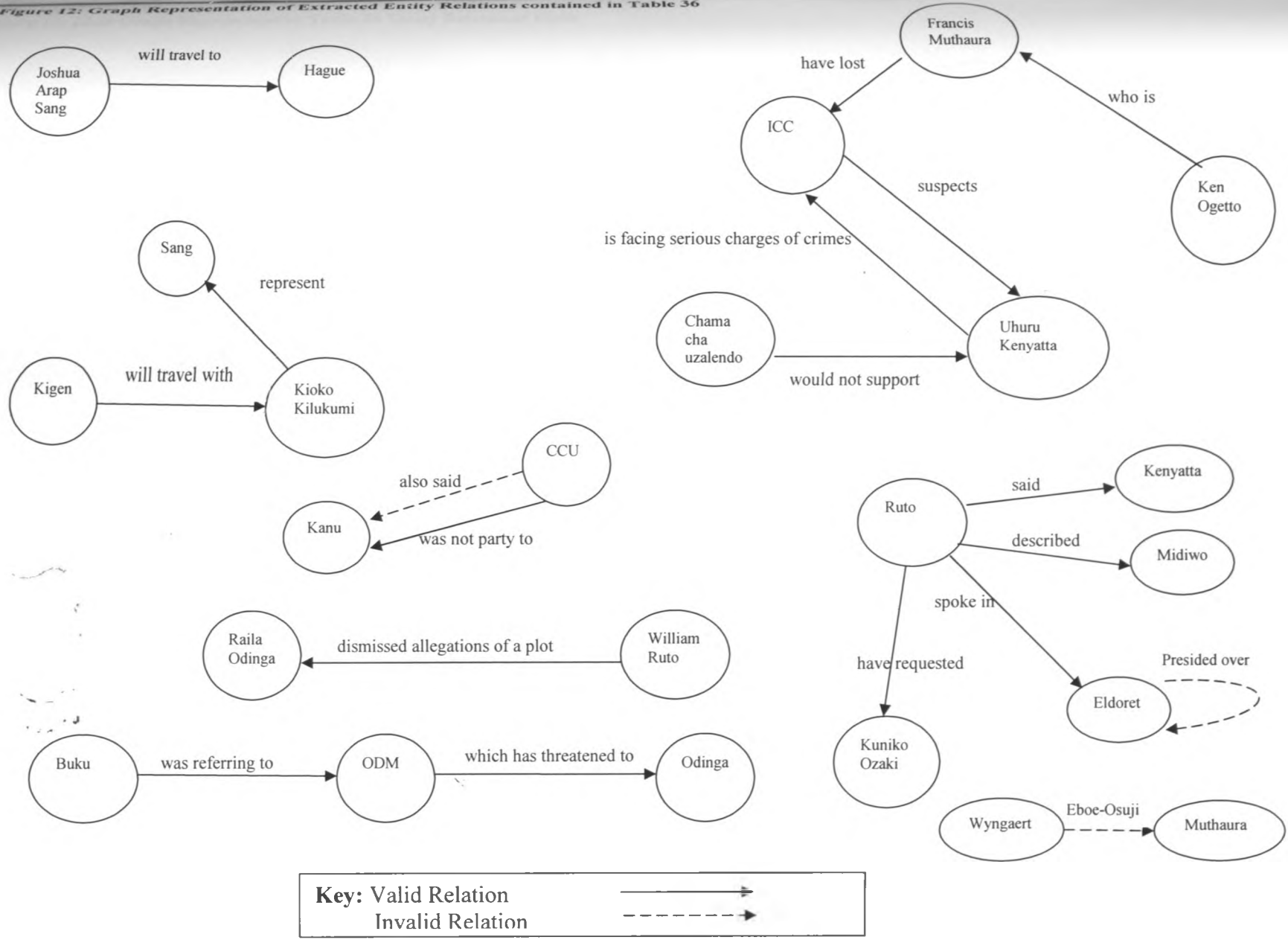
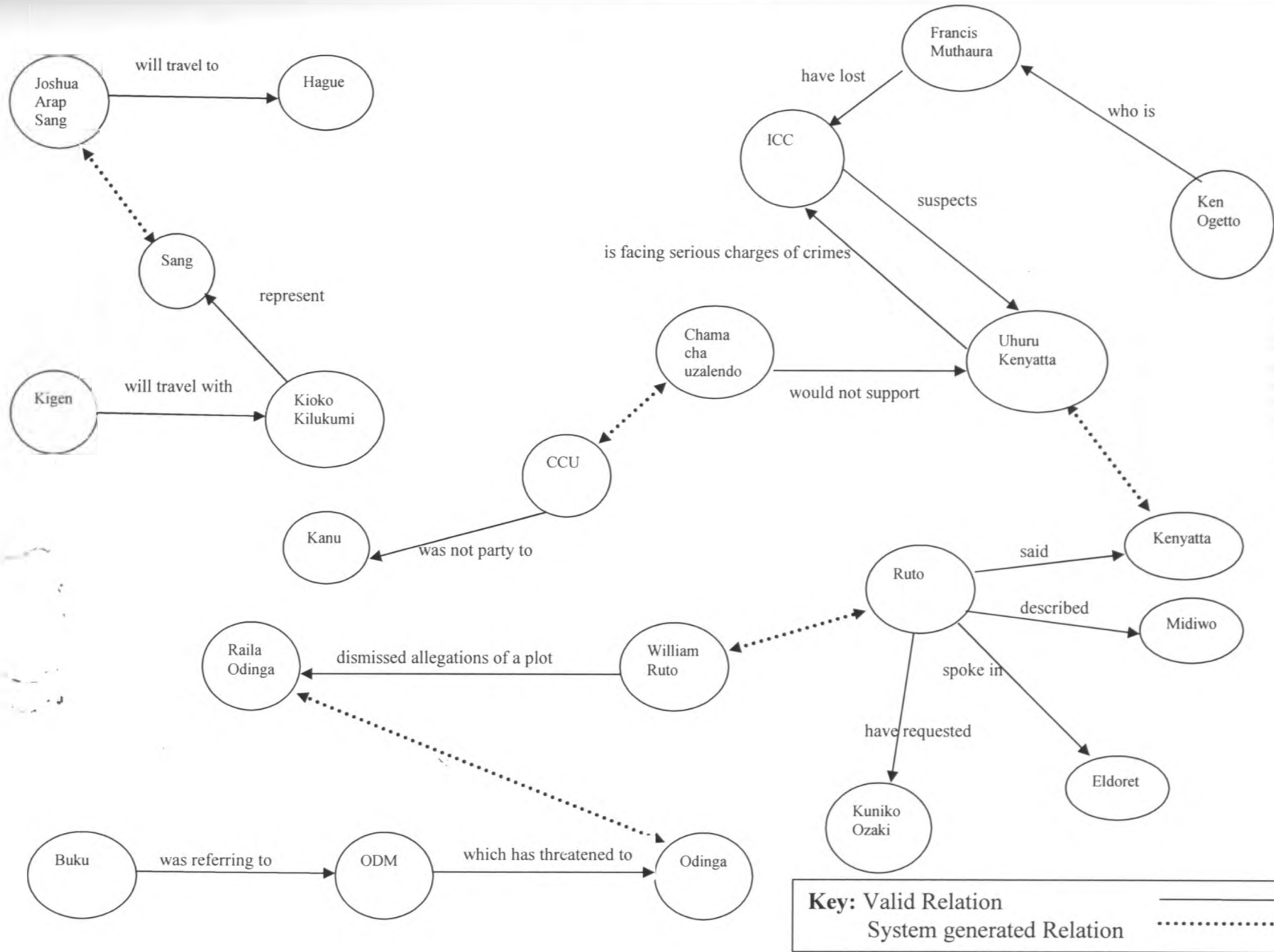


Figure 13: Ideal Graph Representation Table 36 Entity Relational Facts



CHAPTER FIVE: DISCUSSION

5.1. Conclusion

This research presented a Named Entity Relational Facts Characterizer that can be used to find out, how two (2) named entities are related to each other in an unstructured Incident Report text corpus. In this research we sought to answer the question, “What Key issues should be addressed in order to build an Entity Relation Characterizer?”

To answer the research question we conceptualized the research problem as two sub problems; Relation Extraction and Relation Representation. We extended the work of Minkov (2008), which outlines the use of graphs in Personal Information Management (PIM) and Parsed Text domains. Further, using the knowledge provided by Banko (2009), of the existence of a single learning model for the English language (Open Information Extraction) and the tool (REVERB) provided by Fader et al. (2011), we built a prototype Named Entity Relation Characterizer, which we evaluated by conducting experiments on a Test corpus of ten (10) documents.

We conclude that reading and representation of facts contained in Incident Reports can be achieved using Open Information Extraction tools such as ReVerb, in conjunction with a characterizer and graph algorithms. To ensure one builds a fairly accurate Entity Relation Characterizer the following issues should be addressed:

- I. **The Quality of the Text Corpus:** - A professionally authored English text corpus containing wh-patterns is an important requisite in the ultimate success of a characterizer. Since the frequency and distribution of misspelled words over sentences in a document can cause the underlying POS tagger to erroneously identify misspelled words as proper nouns leading to incorrect extraction of relations at the expense of valid relational extractions. See Table 39.
- II. **The Choice of the underlying POS tagger and English dictionary:** - The English dictionary that is chosen for use in POS tagger determines the effectiveness of the system. If the underlying dictionary omits valid English words the POS tagger is likely to tag them as proper nouns, and hence affect the performance of the characterizer. Additionally the effectiveness of the underlying Part of Speech tagger is of great importance, care should be taken to select a POS tagger that is suited for the domain that one is covering.

- III. **The Named Entity Dictionary:** - The size of the named entity dictionary matters, a large dictionary ensure that one can characterize more facts accurately, however one needs to eliminate vernacular names that are spelt in the same way as English words in order to improve the performance of the Characterizer. This process of elimination increases the likelihood of relations involving the eliminated names being omitted reducing the accuracy of the characterizer.
- IV. **Document level named entity resolution mechanism:** - Though the relational extraction sub problem focused primarily on relational facts expressed in a single sentence, for a characterizer to be effective a mechanism to link mentions of the entity across sentences within a document can be implemented. We conclude that the ideal representation of facts contained in Table 36 should be as shown in Figure 12, with the dotted arrows indicating missing relations that a human reader can infer from existing facts contained within the document that they are reading and hence the need of a document level named entity resolution mechanism.

5.2 Limitations

The characterizer uses a greedy algorithm in tagging named entities, this does not always yield an optimal solution; in circumstances where a named entity is composed of two or more tokens e.g. 'Kenya International Conference Center' the characterizer takes the first token 'Kenya' and tags it as Person since the named entity dictionary also identifies 'Kenya' as a Person. To overcome this limitation user's who may have a large number of named entities identifying facilities and part of their names being shared with peoples names could consider preprocessing their documents to have names such as 'Kenya International Conference Center' represented as a single token 'Kenya_International_Conference_Center' prior to having them run through the relational extractor.

The characterizer requires human intervention in resolving co-referential tokens (distinct words referring to the same object or person) e.g. In one a sentence a writer speaks of 'Chama Cha Uzalendo' in the next sentence the writer speaks of 'CCU', which are the initials representing Chama Cha Uzalendo.

5.3 Recommendations for further Research

Further research may be conducted on the use of graph data structures in entity name disambiguation in unstructured text, which can serve as the Document level name entity resolution mechanism highlighted under the conclusion section.

REFERENCES

- (ACE), A. C. E. 2005. English Annotation Guidelines for Relations Version 5.8.3 – 2005.07.01. Available: <http://www ldc.upenn.edu/Projects/ACE/>.
- (NIST), N. I. O. S. A. T. Proceedings of the 6th Message Understanding Conference (MUC-7), 1998.
- AGICHTEIN, E. & CUCERZAN, S. 2005. Predicting Accuracy of Extracting Information from Unstructured Text Collections. *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*.
- AGICHTEIN, E. & GRAVANO, L. 2000. Snowball: Extracting relations from large plain-text collections. *In Proceedings of the Fifth ACM International Conference on Digital Libraries*.
- AKMAJIAN, A., DEMERS, R., FARMER, A. K. & HARNISH, R. M. 2001. An introduction to Language and Communication. fifth ed.: MIT Press.
- ATKINS, S., CLEAR, J. & OSTLER, N. 1992. Corpus design criteria. *Literary and Linguistic Computing*, Vol. 7.
- BAGGA, A. Analyzing the reading comprehension task. NAACL-ANLP 2000 Workshop Syntactic and Semantic Complexity in Natural Language Processing Systems, 2000.
- BANKO, M. 2009. *Open Information Extraction for the Web*. Doctor of Philosophy Dissertation, University of Washington.
- BANKO, M., CAFARELLA, M. J., SODERLAND, S., BROADHEAD, M. & ETZIONI, O. 2007. Open information extraction from the web. *In the Proceedings of the 20th International Joint Conference on Artificial Intelligence*.
- BASS, L., CLEMENTS, P. & KAZMAN, R. 1998. *Software Architecture in Practice*, Addison-Wesley.
- BIBER, D. 1993. Representativeness in Corpus Design. *Literary and Linguistic Computing*. Oxford University Press.
- BRIN, S. Extracting patterns and relations from the World Wide Web. In WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98, 1998.
- COWIE, J. & LEHNERT, W. 1996. Information Extraction. *Communications of the ACM*, Vol. 39, pg 80–91.
- CUNNINGHAM, H. 2000. *Software Architecture for Language Engineering*. Doctor of Philosophy Dissertation, University of Sheffield.
- CUNNINGHAM, H., MAYNARD, D., TABLAN, V., URSU, C. & BONTCHEVA, K. 2010. Developing Language Processing Components with GATE. University of Sheffield.

- ETZIONI, O., BANKO, M. & CAFARELLA, M. J. Machine reading. In Proceedings of the 21st National Conference on Artificial Intelligence, 2006.
- ETZIONI, O., FADER, A., CHRISTENSEN, J., SODERLAND, S. & MAUSAM 2011. Open Information Extraction: the Second Generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*.
- FADER, A., SODERLAND, S. & ETZIONI, O. Identifying Relations for Open Information Extraction. Proceedings of the Conference of Empirical Methods in Natural Language Processing ({EMNLP} '11), July 27-31 2011 Edinburgh, Scotland, UK.
- FREITAG, D. 1998. *Machine Learning for Information Extraction in Informal Domains*. Doctor of Philosophy Dissertation, Carnegie Mellon University.
- GRAMA, A., GUPTA, A., KARYPIS, G. & KUMAR, V. 2003. *Introduction to Parallel Computing*, Addison-Wesley.
- JURAFSKY, D. & MARTIN, J. H. 2003. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Pearson Education.
- KUDENKO, D. & HIRSH, H. Feature-based Learners for Description Logics. Proceedings of the International Workshop on Description Logics (DL99), 1999.
- MCENERY, T., XIAO, R. & ONO, Y. 2006. *Corpus-based Language studies: An advanced Resource Book*, Taylor & Francis.
- MILLS, A. P. 2007. *Knowledge of Language* [Online]. Internet Encyclopedia of Philosophy (IEP). Available: <http://www.iep.utm.edu/knowlang/> [Accessed 05-June 2012].
- MINKOV, E. 2008. *Adaptive Graph Walk Based Similarity Measures in Entity-Relation Graphs*. Doctor of Philosophy Thesis, Carnegie Mellon University.
- MONCECCHI, G., MINEL, J.-L. & WONSEVER, D. 2003. A survey of kernel methods for relation extraction. *Journal of Machine Learning*, vol. 3, 1083-1106.
- PRESSMAN, R. S. 2001. *Software Engineering: A Practioner's Approach*, McGraw-Hill.
- ROBERT, G. & WILKS, Y. 1998. Information Extraction: Beyond Document Retrieval. *Journal of documentation*, 54, 70-105.
- SYAL, P. & JINDAL, D. V. 2007. *An Introduction to Linguistics, Language, Grammar and Semantics*, PHI Learning Pvt. Ltd.

APPENDIX 1: Part of Speech Tags

Alphabetical list of part-of-speech tags used in the Penn Treebank Project

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential <i>there</i>
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP\$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	<i>to</i>
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP\$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Source <http://www.ling.upenn.edu>

APPENDIX 2: News Stories in Test Corpus

Source: http://www.nation.co.ke	Date: 6 th Jun 2012
File Name: Nation-06-6-2012.txt	Author: Oliver Mathenge

Radio presenter Joshua arap Sang and his three lawyers will travel to The Hague on Saturday for a status conference ahead of his trial at the International Criminal Court. According to Mr Sang, lawyers Philemon Koech, Joel Bosek and Katwa Kigen have already applied for their visas and are awaiting clearance from the Dutch Embassy in Nairobi.

Mr Kigen will travel with Mr Kioko Kilukumi with whom they represent Mr Sang's co-accused, Eldoret North MP William Ruto. The lawyers did not say whether Mr Ruto would be travelling but sources close to the politician have disclosed that he would not be going.

The prosecution, the defence, the ICC registry and the victims' lawyers in the case against Mr Ruto and Mr Sang will attend the status conference on Monday where the trial date for the two will be set. Mr Sang and Mr Ruto have requested Trial Chamber judges Kuniko Ozaki, Christine Van den Wyngaert and Chile Eboe-Osuji to have the trial date set after the next General Election. Mr Ken Ogetto, who is in former Head of Civil Service Francis Muthaura's defence team, will also be travelling on Saturday for the status conference in the second case which will be held on Tuesday.

Lawyers for Mr Muthaura's co-accused, Deputy Prime Minister Uhuru Kenyatta, were unavailable for comment but are also expected to travel over the weekend. Sources have said that the two will not travel to The Hague. Both meetings will begin at 2pm Kenyan time but the suspects are not required to be in court in person.

On average, it has taken between six and eight months for previous cases to start after the status conferences. The judges have asked parties to the case to make any submissions regarding the agenda of the status conferences. "If the parties, the legal representatives of victims and the registry are currently aware of any other issue that is required to be resolved before the commencement of the trial, they should bring it to the attention of the Chamber promptly," the Trial Chamber judges said.

Apart from setting the date of trials, the conference will also set the time-lines and format of disclosing evidence including witnesses who will require protection. The prosecution has indicated it will require a year to present its evidence in each of the cases.

Source: http://www.nation.co.ke	Date: 18 th Apr 2012
File Name: Nation-18-04-2012.txt	Author: Fred Mukinda and Samuel Koech

ICC suspects Uhuru Kenyatta and William Ruto on Tuesday dismissed allegations of a plot to assassinate Prime Minister Raila Odinga. Mr Kenyatta's spokesman, Mr Munyori Buku, dismissed the claims, saying that they were part of a "scheme to camouflage the biggest failure in that party (Orange Democratic Movement) - dictatorship and resistance to change". "This issue of dropping names of people holding senior positions recklessly, instead of working for Kenyans, is a scheme that borders on devilish acts," said Mr Buku.

He added: "They should spend more energy resolving their party disputes. There is room for competitive politics without making defamatory statements." Mr Buku was referring to nomination of ODM's presidential candidate, which has threatened to split the party, with some members insisting on unchallenged nomination of Mr Odinga as party leader while those allied to his deputy, Mr Musalia Mudavadi, proposing a competitive process to pick a contender.

Mr Ruto, the Eldoret North MP, said separately that it was unfortunate that people can make casual statements on sensitive matters pertaining to security. He urged Kenyans to treat ODM chief whip Jakoyo Midiwo's claims that there was a plot to kill Mr Odinga with the "contempt they deserve". "It is so unfortunate that people can make alarming, unsubstantiated statements that have no basis and are bound to cause unnecessary friction, especially at this time when the country is gearing for the general election," said Mr Ruto.

Mr Ruto said Deputy Prime Minister Kenyatta, Foreign Affairs minister Sam Ongeru and himself had recorded statements with the police and urged security agencies to move with haste in investigating the motive behind the claims. Mr Ruto described Mr Midiwo's claims as a "useless story", adding that it is only a mad man who can think of hatching a plot to assassinate the Prime Minister. Mr Ruto spoke in Eldoret yesterday when he presided over the Eldoret West District education and prize giving day. "If the PM can be at risk with the heavy security detail attached to him, then what about the ordinary Kenyan citizen?" asked Mr Ruto.

Source: Nation Newspaper	Date: 17 th Mar 2012
File Name: Nation-17-3-2012.txt	Author: -
<p>Chama Cha Uzalendo yesterday said that it would not support Mr Uhuru Kenyatta's presidential bid. In a statement signed by Spokesman Johnson Ngari CCU also said it was not party to an agreement reportedly signed by 15 parties to form an alliance with Kanu. "CCU is aware that Uhuru Kenyatta is facing serious charges of crimes against humanity at the ICC and it will therefore, be naive for the party to enter into an alliance with such a person unless fully cleared".</p>	

Source: The Star newspaper	Date: 23 rd June 2012
File Name: Star-23-06-2012.txt	Author: Nzau Musau
<p>Deputy Prime Minister Uhuru Kenyatta and former head of civil service Francis Muthaura have lost a bid to gag the ICC prosecutor from contacting their witnesses. Trial judges Kuniko Ozaki (presiding judge), Christine Van den Wyngaert and Eboe-Osuji rejected the application initially sought by Muthaura's lawyer Karim Khan and later supported by Uhuru.</p> <p>The two are facing trial at the court over crimes against humanity. They are charged alongside Eldoret North MP William Ruto and radio presenter Joshua arap Sang. Khan had during the status conference held on June 12, applied to the court issue an interim order prohibiting the prosecution now led by Fatou Bensouda from contacting potential witnesses until the judges issue a ruling on the procedure of contacting witnesses.</p>	

Source: http://www.nation.co.ke	Date: 13 th Jun 2012
File Name: Nation-13-06-2012.txt	Author: Nation Correspondent
<p>Rift Valley MPs allied to URP leader William Ruto and those supporting Prime Minister Raila Odinga yesterday held separate meetings in what one faction said will culminate in a joint rally.</p> <p>Assistant minister Langat Magerer who has been a strong supporter of the PM in the province said he was ready to work with Mr Ruto's URP. Briefing the press after a meeting attended by Roads minister Franklin Bett, assistant minister Julius Murgor and MP Joyce Laboso, the Raila allies said they were brainstorming on how different parties in the province can work together.</p> <p>"We are ready to work with parties such as Mr Ruto's URP and a meeting will be held in Nakuru this month to chart the way forward for the region," Mr Magerer said.</p> <p>The MP, who is the coordinator of the meeting, said he had invited Kanu, Wiper, PNU, TNA, UDF, UDM and the National Vision Party. "I have sent out a notification to all of them," Mr</p>	

Source: http://www.nation.co.ke	Date: 13 th Jun 2012
File Name: Nation-13-06-2012.txt	Author: Nation Correspondent
<p>Magerer said. MPs allied to Mr Ruto were on the other hand locked up in a closed-door meeting for more than two hours.</p> <p>But briefing the press after the meeting in Nairobi, Belgut MP Charles Keter said they were meeting to push some URP rallies to next week to pave way for the burial of the late Internal Security minister George Saitoti and his assistant Orwa Ojodeh.</p>	

Source: Standard Newspaper	Date: 17 th Feb 2012
File Name: Standard-17-02-2012.txt	Author: Leonard Korir
<p>Five Poachers were arrested at a bridge near Kichwa Tembo Lodge in Masai Mara Game Reserve and an assortment of weapons recovered. Mara Conservancy rangers based at OIOLoolo gate who were on a night patrol acted on a tip off from a community scout and laid a trap, netting the men as they crossed the bridge.</p> <p>The rangers led by warden Joshua ole Naiguran, arrested the poachers and recovered an AK47 and G3 rifles and 66 rounds of ammunititon. Naiguran said the poachers were from Lolgorian area where they hired guns and were headed for an operation in Aitong area of Narok South. One of the criminals escaped. Naiguran said the criminals are part of a dreaded poaching gang responsible for the recent massive killings of elephants in Sitoka, Lolgorian, Olosetu and Laila forest areas in Trans Mara West District. More than 28 elephants have reportedly been killed in the last five months outside the reserve.</p> <p>Mara Conservancy Chief Executive Officer, Brian Heath, said following intensified 24-hour security patrols around the game reserve, poachers were now shifting base to Narok. He said they were working with security officers from Narok and Tanzania to wipe out the gangs. "We are not taking any chance we have reinforced our security personel to deal with these illegal groupings whose activities undermine tourism and wildlife heritage", said Heath. The incident comes a fortnight after revelations of an ongoing illegal trade of elephant tusks.</p>	

Source: http://www.the-star.co.ke	Date: 2 nd Mar 2012
File Name: Star-02-03-2012.txt	Author: Hussein Salesa
<p>MORE than 20 elephants have been killed in the vast Marsabit Forest in the Central division of Marsabit county in the last two months as Ethiopian poachers invade the forest. Marsabit Central DC Ruto Kipchumba said hundreds of poachers have found their way into the forest from Ethiopia and are causing havoc by killing elephants.</p> <p>He said the poachers mutilate and remove the ears and private parts of the elephants before killing them and taking away the trophies. Kipchumba said the poachers collude with the locals and invade the vast Marsabit forest where they are also terrorise residents. The DC called upon the Kenya Wildlife Service to urgently intervene and support security officers in patrolling the vast Marsabit county and flush out the militias. He said efforts by the police to step up the fight against poaching in the area are being hampered by the vast and thick forest.</p> <p>Hundreds of poachers from Somalia and Ethiopia have invaded the region to indulge in poaching due to readily available market in other parts of the country. Kenya Wildlife Service has been asked to step up operations against the poachers in the Northern frontier circuit which comprises Mt Marsabit and Kenya forest, Isiolo, Meru, Marsabit and Laikipia districts. Upper Eastern regional commissioner Isaih Nakoru said the government will not tolerate poaching in the region. Security officials in Marsabit said more than 25 elephants are suspected to have been killed by armed poachers along the elephant corridors in the region in the last one month.</p>	

Source: http://www.the-star.co.ke	Source: 12 th Mar 2012
File Name: Star-12-03-2012.txt	Author: Kirimi Murithi
<p>Kenya Wildlife Service rangers guarding the Meru National Park have killed a suspected poacher and recovered a G3 rifle and 56 bullets. The suspect, who was in gang of four, was shot dead at the weekend in a fire exchange with the the rangers who were on patrol on the northern conservation area. Picha Lokitela , the Meru conservation area KWS commander, said the other poachers escaped. "We were patrolling Rapsugul area at around eleven in the morning when the gang fired at them," he said. "Poaching activities are rampant in this area in the evening. The poachers hide in the grasslands," said Lokitela. "This gang has been operating in the park for a long time, eluding security officers. We will pursue those who escaped and bring them to book."</p>	

Kenya Wildlife Service rangers in Tsavo East National park have killed three suspected poachers. The three suspects were gunned down after a shootout at Batalita in the northern part of Tsavo East National Park on Monday evening. One of the suspects fled with gunshot wounds. Sources at the KWS revealed that three AK-47 rifles and 100 bullets were recovered during the attack.

Assistant Director in charge of Tsavo conservation Wilson Korir said the rangers were on a routine patrol in the park on Monday morning when they noticed footprints which raised suspicions poachers could be in the park. He said the rangers traced the footprints from morning till evening when they encountered the suspects and a shootout ensued. "Our officers tracked down the suspects for the whole day before they stumbled on them and gunned down three of them. We are still in hot pursuit of the one who fled," the director said.

Korir called on the public to report to police any suspected person seeking medical attention as they continue with investigations. "The manhunt is still on and we shall not sleep until the suspect is brought to book," Koriri said. The incident comes barely a month after two KWS officers were killed at Sagalla ranch where the assailants made away with two guns and several bullets of the rangers.

Recently, two elephants were killed by suspected poachers at Lwalenyi ranch in Mwatate district. The killings raised concerns among conservationists over the safety of wildlife in the area. Unconfirmed reports indicated that in the last seven months, more than 30 elephants have been killed in private ranches in Tsavo.

Korir, however ruled out any possibilities that the suspected poachers could be the same who had killed the KWS rangers adding that "the incidents happened in very different areas which are very far apart,". He said that they were still in hot pursuit of the suspects adding that the killing of the poachers was a major breakthrough in the fight against poaching that was threatening the elephant and rhino population in the country.

Taita OCPD Nathaniel Aseneka said that a contingent of police had been dispatched to the scene to reinforce security. "Our officers spent a night at the scene guarding the bodies and the recovered weapons until yesterday morning as other officers mounted a search of the runaway suspect,"said the police boss. By yesterday morning, plans were still underway to remove the bodies to the Voi district hospital mortuary.

Source: <http://www.the-star.co.ke>

Source: 16th Mar 2012

File Name: Star-16-3-2012.txt

Author: Mosoku Geoffrey

Fifteen small parties have now announced that they will form a coalition with Deputy Prime Minister Uhuru Kenyatta's Kanu. The parties say the decision was reached after long period of consultations between them and Kanu. "We have looked around among all presidential candidates and resolved to support Uhuru and enter into a coalition with Kanu," Onyango Oloo of Democracy and Freedom Party said.

He said the parties were informed by the constitutional provision which allowed for coalitions before or after elections. "We have signed an agreement which is binding us tighter and we'll be approaching the campaign jointly," Stephen Nyarangi of the People's Democratic Party said. They were addressing a press conference yesterday evening at a city hotel after a day-long meeting with Kanu organising secretary Justin Muturi.

Chama cha Mwananchi, Party of Hope, Kenda, National Patriotic Party, Chama Cha Uzalendo, People's Party of Kenya and New Revival Generation are some of the parties who have endorsed Uhuru's candidature. At the same time, the parties told off Prime Minister Raila Odinga over assertions that Uhuru and Eldoret North MP William Ruto should be in jail.

They said the PM would rather other candidates were in jail than face him in the polls. "The Prime Minister's stature does not allow him to make such 'reckless' statements to the effect that his competitors should be behind bars," Oloo said. "He should be ready to face them in the polls if he is a democrat that he has been calling himself over the years." He said that the parties will soon make public details of their coalition agreement and will be seeking to get more parties on board. The parties said they support a December election date and not March 2013 as proposed by President Kibaki.


```

lst =[]
mylst=[]
sentenceList=[] #This variable contains the list of sentences that are contained in the open file
testlist =[]
testlist2 =[]
toolkitLemmatizer = nltk.WordNetLemmatizer()

cnxn = pyodbc.connect('DSN=test;') #Connect to a data source named test
cursor = cnxn.cursor()
class InputSentence:
    def __init__(self,variableList):
        self.fileName = variableList[0]
        self.sentenceNo= variableList[1]
        self.argument1 =variableList[2]
        self.relationPharse = variableList[3]
        self.argument2 = variableList[4]
        self.arg1StartIndex = variableList[5]
        self.arg1EndIndex = variableList[6]
        self.relationStartIndex = variableList[7]
        self.relationEndIndex = variableList[8]
        self.arg2StartIndex = variableList[9]
        self.arg2EndIndex = variableList[10]
        self.extractionConfidence = variableList[11]
        self.originalSentence =variableList[12]
        self.posTags = variableList[13]
        self.chunkTags = variableList[14]
        self.normalizedArg1 = variableList[15]
        self.normalizedRelation = variableList[16]
        self.normalizedArg2 = variableList[17]

def searchChunkTags():
    index = 0
    str =''
    new_sentence = ''
    startIndex = 0
    endIndex =0
    while index < len(chunkList):
        print chunkList[index], originalSentence[index], index

        if (index == relationalPhraseStartIndex):
            new_sentence = new_sentence.strip() + '+' '<REL:>'

        if (index == relationalPhraseEndIndex):
            new_sentence = new_sentence.strip() + '+' '</REL:>'

        if (chunkList[index] =='B-NP'):
            startIndex = index
            endIndex = 0

        if (chunkList[index] =='B-NP')and (chunkList[index + 1] <>'I-NP'):
            endIndex = index

        if (chunkList[index] =='B-NP')and (chunkList[index + 1] =='I-NP'):
            m = index
            while m < len(chunkList):#-1:
                if (chunkList[m] =='I-NP') and (chunkList[m + 1] <>'I-NP') :
                    endIndex =m
                    break

```



```
m = m + 1
```

```
if (chunkList[index] == 'B-NP'):
```

```
    str = "" # initialize the string variable
```

```
    k = startIndex
```

```
    word = ""
```

```
    suggestWord = ""
```

```
    while k < (endIndex+1): # This takes care of boundary conditions
```

```
        checkdict = originalSentence[k].lower()
```

```
        if len(checkdict) > 1:
```

```
            if checkdict[0] in punctuation:
```

```
                checkdict = checkdict[1:len(checkdict)]
```

```
            if checkdict[len(checkdict)-1] in punctuation
```

```
                checkdict = checkdict[0:len(checkdict)-2]
```

```
        checkdict = checkdict.rstrip()
```

```
        if checkdict.endswith('s'): # remove possessives from names
```

```
            checkdict = checkdict[0:len(checkdict)-2]
```

```
        checkdict = checkdict.strip()
```

```
        word = repr(dict.get(checkdict.strip()))
```

```
        word = word.replace("'", "") # remove the unicode ending
```

```
        if word <> 'None':
```

```
            word = word[1:len(word)]
```

```
            str = str.strip() + '<' + word + '>' + checkdict.title() + '</' +
```

```
word + '>'
```

```
            suggestWord = ""
```

```
        else:
```

```
            str = str.strip() + ' ' + checkdict
```

```
            if checkdict <> "" and checkdict not in punctuation and not
```

```
wordnet.synsets(checkdict) and checkdict.isdigit() == False and checkdict.lower() not in
```

```
stopwords.words('english'):
```

```
                suggestWord = suggestWord.lstrip() + ' ' + checkdict
```

```
        checkdict = str.lower()
```

```
        word = repr(dict.get(checkdict.strip()))
```

```
        word = word.replace("'", "")
```

```
        if word <> 'None': # strip leading whitespace
```

```
            word = word[1:len(word)]
```

```
            str = '<' + word + '>' + str.title() + '</' + word + '>'
```

```
        suggestWord = suggestWord.lstrip()
```

```
        if suggestWord.endswith('s') or suggestWord.endswith('\s'):
```

```
            suggestWord = suggestWord[0:len(suggestWord)-2]
```

```
        if k == endIndex and word == 'None' and suggestWord <> ":
```

```
            cursor.execute("select EntityName from suggestedNE where
```

```
EntityName = ?", suggestWord)
```

```
            row = cursor.fetchone()
```

```
            if row == None:
```

```
                cursor.execute("insert into suggestedNE(EntityName,
```

```
FileName) values (?,?)", suggestWord, fileName)
```

```

                                cnxn.commit()
                                k = k + 1
                                new_sentence = new_sentence.strip() +' '+ str.strip()

                                str=""

                                if (chunkList[index] <>'B-NP') and (chunkList[index] <>'I-NP'):
                                    #print chunkList[index]
                                    new_sentence = new_sentence.strip() +' '+ originalSentence[index].strip()

                                index = index+1
                                new_sentence = re.sub( r' </PER:> <PER:> ', '', new_sentence)
                                characterizeRelation(fileName,sentenceNo,originalSentence,new_sentence)

def characterizeRelation(fileName, sentenceNo, originalSentence, new_sentence):
    sample = new_sentence.split() #Split the new sentence into a list
    entityRelation=[]
    startIndex = 0
    endIndex = 0
    k = 0
    str=""
    while(k < len(sample)):
        if sample[k].startswith('<',0,1)==True and sample[k].startswith('<',0,2)==False:
            startIndex = k
        if sample[k].startswith('<',0,2)==True: #Check for the end of entity tag
            endIndex = k
        if startIndex < endIndex and endIndex < 0:
            j=startIndex
            while(j < endIndex+1):
                str = str +' '+ sample[j]
                j=j+1
            entityRelation.append(str)#Append the extracted entity to a list of Entities and
Relations
                                startIndex = 0
                                #initialize the start and end index after
processing
                                endIndex = 0
                                str=""
                                k =k+1

                                m = 0
                                #print entityRelation
                                while(m < len(entityRelation)-1):
                                    if entityRelation[m].startswith(' <REL:>', 0,8)==False: #Focus on processing Entities
alone
                                        entityName = entityRelation[m]
                                        entityType = entityName[2:5]
                                        #Extract the Entity Type i.e.PER,
ORG,
                                        entityName = entityName[8:len(entityName)-8]
                                        #Extract the Entity itself

                                        cursor.execute("select EntityName from InvertedIndex where EntityName = ? ",
entityName )
                                        row = cursor.fetchone()
                                        if row==None: #Check to see if the Entity is in the
Inverted Index if not insert it
                                            cursor.execute("insert into InvertedIndex(EntityType, EntityName)
values (?, ?)",entityType, entityName )
                                            cnxn.commit()

```

```

entityName )
        cursor.execute("select IndexId from InvertedIndex where EntityName = ? ",
        row = cursor.fetchone()
        if row <> None:
            indexId = row[0]

        cursor.execute("select FileName,SentenceNo,IndexId from EntityMetaData
where FileName =? and SentenceNo =? and IndexId =?", fileName, sentenceNo,indexId)
        row = cursor.fetchone()
        if row==None:
            cursor.execute("insert into EntityMetaData(FileName, SentenceNo,
OriginalSentence,IndexId) values (?,?,,?)", fileName, sentenceNo, new_sentence,indexId)
            cnxn.commit()
        else:
            if m>0:
                print entityRelation[m-1], entityRelation[m], entityRelation[m + 1]
                relation = entityRelation[m]
                relation = relation[8:len(relation)-8]

                entityName = entityRelation[m-1]
                entityType = entityName[2:5]
                entityName = entityName[8:len(entityName)-8]

                cursor.execute("select IndexId,EntityName from InvertedIndex where
EntityName = ? ", entityName ) #Obtain the Entity Index of the Entity
                row = cursor.fetchone()
                primaryIndexId = row[0]
                primaryEntityName=row[1]

                entityName = entityRelation[m+1]
                entityType = entityName[2:5]
                entityName = entityName[8:len(entityName)-8]

                cursor.execute("select IndexId from InvertedIndex where EntityName
= ? ", entityName )
                row = cursor.fetchone()
                if row==None:
                    cursor.execute("insert into InvertedIndex(EntityType,
EntityName) values (?, ?)",entityType, entityName )
                    cnxn.commit()

                cursor.execute("select IndexId,EntityName from InvertedIndex where
EntityName = ? ", entityName ) #Obtain the Entity Index of the Entity
                row = cursor.fetchone()
                secondaryIndexId = row[0]
                secondaryEntityName = row[1]

                #Check if there is similar Entity Relation in the EntityRelation Table
                cursor.execute("select PrimaryEntityindex, Relation,
SecondaryEntityIndex from ExtractedRelation where PrimaryEntityindex = ? and Relation = ? and
SecondaryEntityIndex =?", primaryIndexId, relation, secondaryIndexId)
                row = cursor.fetchone()
                if row==None:
                    cursor.execute("insert into
ExtractedRelation(PrimaryEntityindex, Relation, SecondaryEntityIndex, PrimaryEntityName,
SecondaryEntityName,fileName) values (?,?,,?,?)", primaryIndexId, relation,
secondaryIndexId,primaryEntityName,secondaryEntityName,fileName)
                    cnxn.commit()

```

m = m+1

```

dict = {}

rootdir = sys.argv[1]

for root, subFolders, files in os.walk(rootdir):
    for file in files:
        filePath = rootdir + '\\' + file
        testReader = csv.reader(open(filePath, 'rb'))

        for row in testReader:
            str = ', '.join(row) #converts each row that is read from a list into a
string
            mylst.append(str.split('\t'))

        for item in mylst:
            new_sentence = InputSentence(item)
            sentenceList.append(new_sentence)

        for i in range(len(sentenceList)):
            str = ".join(sentenceList[i].originalSentence)
            sentenceList[i].originalSentence=[]
            newlist = str.split(' ')
            for item in newlist:
                sentenceList[i].originalSentence.append(item)

            str=""
            newlist=[]
            str = ".join(sentenceList[i].chunkTags)
            sentenceList[i].chunkTags=[]
            newlist = str.split(' ')

            for item in newlist:
                sentenceList[i].chunkTags.append(item)

        cursor.execute("select EntityName, EntityType from NamedEntity")
        rows = cursor.fetchall()
        for row in rows:
            dictkey = row.EntityName.lower()
            dict [dictkey.strip()]= row.EntityType.strip()

        for i in range(len(sentenceList)):
            fileName = sentenceList[i].fileName
            sentenceNo = sentenceList[i].sentenceNo
            chunkList = sentenceList[i].chunkTags
            originalSentence = sentenceList[i].originalSentence
            relationalPhraseStartIndex = int(sentenceList[i].relationStartIndex)
            relationalPhraseEndIndex = int(sentenceList[i].relationEndIndex)
            searchChunkTags()

```

RelationQueryEngine.py

This program is used after the *TagAndCharacterize.py* has run to completion. When a user clicks on the 'Search' button on the *Entity Relation Search* form, after having specified Entity Name and Entity Name 2. A connection ('test') to the database through ODBC is established. Create a

graph structure mygraph containing all the nodes with primary entity. Use the findpath function to find an interlinkage between two entities.

```

import pyodbc
import sys
cnxn = pyodbc.connect('DSN=test;') #Connect to a data source named test
cursor = cnxn.cursor()
mygraph={}
nodelist=[]
cursor.execute("select IndexId from InvertedIndex")
rows = cursor.fetchall()

for row in rows:
    cursor.execute("select SecondaryEntityIndex from ExtractedRelation where PrimaryEntityindex =
?", row.IndexId )
    targetnodes= cursor.fetchall()
    if targetnodes <> None:
        for node in targetnodes:
            nodelist.append(node[0])
            mygraph [row.IndexId]= nodelist
            nodelist=[]

dict={} # Create a dictionary for returning people names given an Index
cursor.execute("select IndexId, EntityName from InvertedIndex")
rows = cursor.fetchall()
for row in rows:
    dict [row.IndexId]= row.EntityName

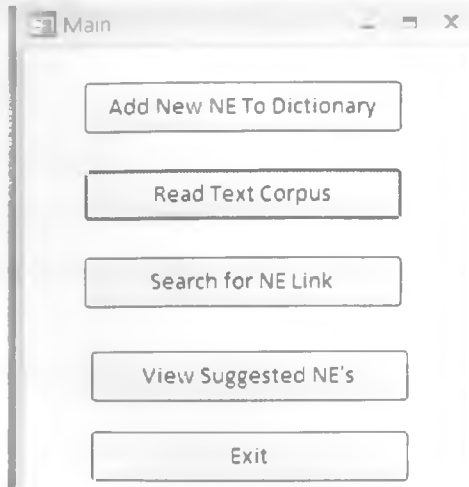
def find_path(graph, start, end, path=[]):
    path = path + [start]
    if start == end:
        return path
    if not graph.has_key(start):
        return None
    for node in graph[start]:
        if node not in path:
            newpath = find_path(graph, node, end, path)
            if newpath: return newpath
    return None

if (len(sys.argv) > 1):
    arg1 = int(sys.argv[1])
    arg2 = int(sys.argv[2])
    linkpath= find_path(mygraph, arg1, arg2)
    if linkpath <> None:
        index =0
        while index < len(linkpath)-1:
            cursor.execute("select PrimaryEntityindex, Relation,
SecondaryEntityIndex,fileName from ExtractedRelation where PrimaryEntityindex = ? and
SecondaryEntityIndex= ?", linkpath[index], linkpath[index+1] )
            rows = cursor.fetchall()
            for row in rows:
                mystr = str(dict.get(row[0])) + "," + str(row[1]) + "," +
str(dict.get(row[2])) + "," + str(row[3])
                print mystr
                index = index + 1
    else:
        print "##No Link, ,,"

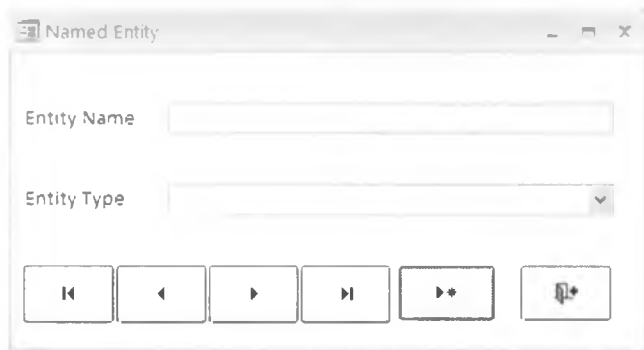
```

APPENDIX 4: User Manual

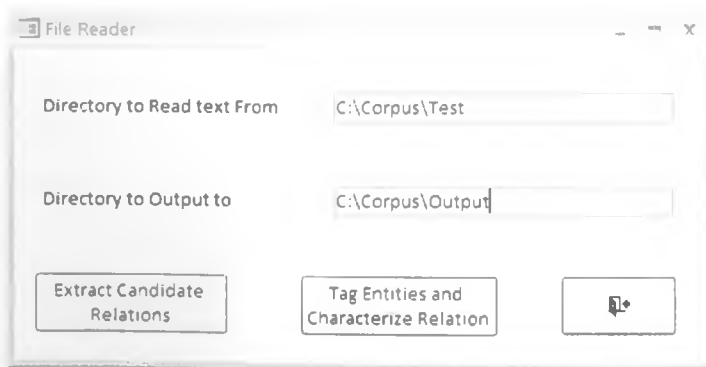
1. To install the Prototype system, follow the steps outlined in section 3.2.4.
2. To run the prototype double click on the MS Access file named *RelationExtraction.mdb*, the main menu appears as shown.



3. To add a new named Entity select "*Add New NE To Dictionary*" from the Main Menu. The window shown below appears

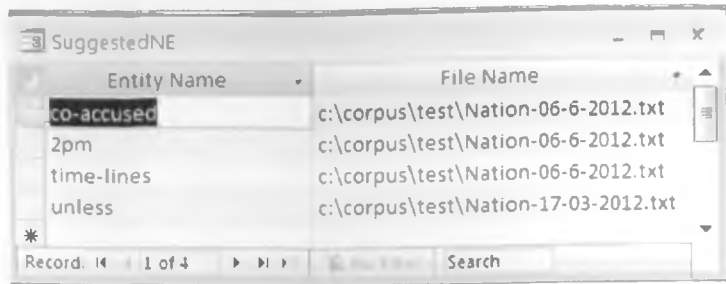


4. To read a text corpus Select "*Read Text Corpus*" from the Main Menu. The window shown below appears.

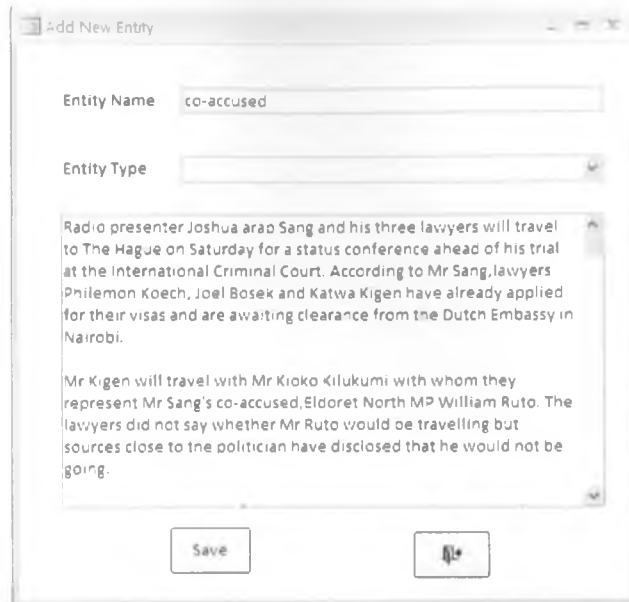


Specify the Input and Output directories. The input directory specifies the location of the Test Corpus, whereas the Output directory specifies the location that the candidate extractions from the Relational Extractor will be stored.

5. To view the suggested Named Entities select “View Suggested NE's” from the Main Menu. The window shown below appears



From the list of suggested Named Entities select a row containing the word that you wish to include in the Named Entities and double the window shown below appears.



6. From the Main Menu shown in Figure 7, select “Search for NE Link”. The window shown below appears. Select the Primary Entity Name and the Secondary Entity Name and then click on Search.

