Named Entity Extraction From Disaster Reports

By

Alice Irankunda (Alice.irankunda@aims.ac.rw)
Supervised by Dr.Yabebal FANTAYE

June 2017

AN ESSAY PRESENTED TO AIMS RWANDA IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE IN MATHEMATICAL SCIENCES

DECLARATION

This work was carried out at AIMS Rwanda in partial fulfilment of the requirements for a Master of Science Degree. I hereby declare that except where due acknowledgement is made, this work has never been presented wholly or in part for the award of a degree at AIMS Rwanda or any other University.

Student: Alice IRANKUNDA 

Supervisor: Yabebal Fantaye 

Co-supervisor: Xavier Vollenwein

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to the Almighty God for His protection and helping me through this program successfully. I would like to express my heartfelt appreciation to my supervisor; Dr. Yabebal Fantaye and co-supervisor Dr Xavier Vollenweider who gave me the necessary support and assistance which propel me to complete this research. I would also like to say a big thanks to my tutor Dr Jan Hązła for his immense contribution towards making this research a success. To my family and loved ones who contributed in diverse ways to my success in this program, i say a big thank you and may God bless you. My final thanks goes to my colleagues, tutors and the administrative staff, through all the challenges you still remained supportive.

Abstract

Reports are key source of information for all activities within organizations. Electronic reports are generated day to day in an unstructured way. It is still a big challenge to know automatically what the reports are talking about. For big organizations like International Federation of Red Cross (IFRC) which work in humanitarian domain, some information from their reports are very important. Automation of extracting information saves time and increases quality. In this research, we are concerned with extracting specific pieces of information called named entities, such as names of persons who participated in IFRC activities, locations, organizations, budget, etc. We used machine learning algorithms such as Stanford NER, Polyglot and Natural Language ToolKit to extract named entities from IFRC reports. We were looking for the answer of "Who did what, when and how?" from the reports.

Contents

1. Literature Review

In today's life, many organizations are generating unstructured data while they are communicating. There are plenty of entities to be extracted. In this research, all reports we considered are written in English.

To identify boundaries of sentences is one of the important prerequisite steps in Natural Language Processing. The punctuation marks cause some ambiguity (Baluja et al., 2000) in texts processing. For example, it is challenging to differentiate the point in abbreviations and a full stop.

There are different approaches to extract entity from documents. Some information extractor (IE) systems are based on regular expressions rules and patterns between words. Other algorithms are based on machine learning concepts.

Named Entity Recognition and Classification (NERC) is one or machine learning algorithms which uses Markov Hidden Model to classify a document.

Due to various forms of contexts and forms various forms of documents, it is not preferable to extract entities manually. Machine learning algorithms give nice and robust results.

Parse tree is a graphical ordered representation of words compose sentences. It bases on rules of phase structure grammar.

1.1 Parse Tree

One of the sentences that compose our sample report says: "Assessment reports indicated 117 deaths, 544 people injured, 12,794 homes damaged and 7,384 houses destroyed", Suppose that this sentence is called "S".

There are two mains steps which are performed to get the entities from this sentence:

- **Tokenizing**: This is a procedure of taking a sentence and extracting the composing atomic linguistic elements i.e. words, verbs, punctuations, adjectives etc . S has the following tokens: ['Assessment', 'reports', 'indicated', '117', 'deaths', ',', '544', 'people', 'injured', ',', '12,794', 'homes', 'damaged', 'and', '7,384', 'houses', 'destroyed']
- **POS**: part-of-speech is a process of attaching to every linguistic element of the sentence a corresponding tag based on grammar rules. The POS of S are: [('Assessment', 'JJ'), ('reports', 'NNS'), ('indicated', 'VBD'), ('117', 'CD'), ('deaths', 'NNS'), (',', ','), ('544', 'CD'), ('people', 'NNS'), ('injured', 'VBN'), (',', ','), ('12,794', 'CD'), ('homes', 'NNS'), ('damaged', 'VBN'), ('and', 'CC'), ('7,384', 'CD'), ('houses', 'NNS'), ('destroyed', 'VBD')]

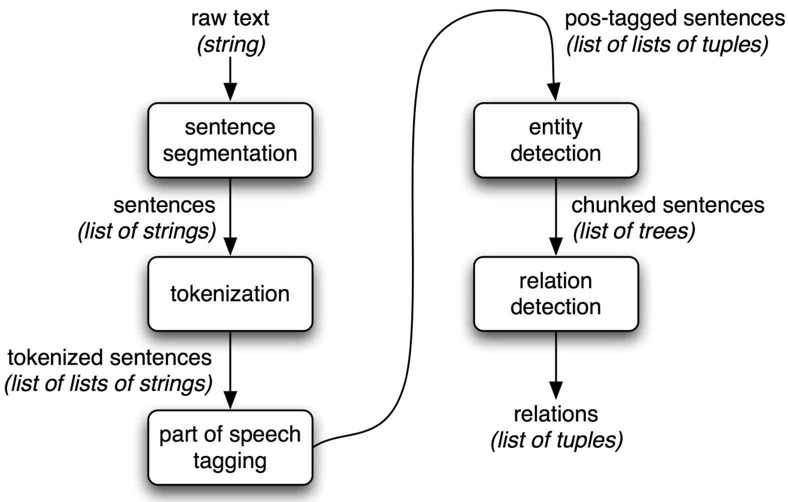
The meanings of the used tags for S:

- JJ: **Adjective**: 'Assessment'.

- NNS: **Noun, plural**: 'reports', 'deaths', 'people','houses'.
- VBD: **Verbs, past tense**: 'indicated', 'injured', 'damaged', 'destroyed'.
- CD: **Cardinal Number**: '117', '544' '12,794','7,384'.
- CC: **Coordinate Conjugation**: 'and'

The parse tree is formed based on the POS. Classification and arrangement of words in a sentence determine the relationship between words.

Figure 1.1: General Extraction Process (Bird et al., 2009)



1.2 Named Entity Recognation and Classification (NERC)

The term "Named entity" has been coined in 1996 by (Grishman and Sundheim, 1996). Entity can be referred as a task, the entity is "named" when it is restricted to one or many rigid designators (Sharnagat, 2014) such as persons, locations, product etc.

Based on the classification of Standard Generalizes Markup Language (SGML) a task can be divided into three subtasks:

1. ENAMEX: locations, products, organizations
2. NUMEX : percentage, quantity
3. TIMEX : time, date

During a process of classification, Capitalization can be used in different ways such as the beginning of the proper noun, the abbreviation, the post of high level profile people etc.

Considering the English language text, if we are given a particular token it is not by chance to determine whether it is a name or not. Some of the approaches to indicate a name are to use capitalization detection of sentence boundaries and dictionaries (Baluja et al., 2000).

To handle this ambiguity, some systems use the special purpose-regular expression grammar, exception rule method, and so on. David Palmer and Marti Hearst in (Palmer and Hearst, 1994) worked on punctuation marks and capitalization of words. They developed an efficient system with high accuracy in automatic sentences boundaries labelling by using the feed forwarding neural-networks where the input was the Part-of-speech (POS) probabilities of all tokens which are surrounding the punctuation. Their output was found as the label of tokens. David Palmer and Marti Hearst 's work was able to find correctly 98.5% for punctuation of sentence-boundaries. A proposed new approach was how to represent the context of punctuation marks without ambiguities.

For extracting entities in a report there are different models which can be used like Hidden Markov model, Supporting Vector Machine (SVM), etc.

1.3 Hidden Markov Model (HMM)

It is a statistical Markov model which trains randomly systems and assumes that future states depend on current trained states. HMM is specifically used for extraction of patterns in texts and speeches. This model is based on Bayesian probability inference which has been initiated in 18th century. HMM is the earliest applied model for Natural Entities Recognition for English language.The way to perform these tasks is to find the most likely sequence of tagged names TN given a sequence of words "SW".

$$P(TN|SW) = \frac{P(SW|TN)P(TN)}{P(SW)} \tag{1.3.1}$$

The equation (1.3.1) is conditional probability, $P(TN|SW)$ can be called posterior and it is the probability of an event tagged names occurring given sequence of word has observed. $P(SW|TN)$ is also called likelihood e.i. it is the probability of observing the sequence of words SW when the given hypothesis tagged name TN is true. On the other handP(TN) doesn't depend on the evidences, P(TN) is called prior e.i. that it is true even if there is no given evidence at all. We can be ignore P(SW) and the remaining objective is to maximise the probability of getting the sequence of tagged names when sequence of words is given.

$$Max\left[P(TN|SW)\right] \tag{1.3.2}$$

Due to assumption that the probabilities of tags are independent from each other, from maximization equation (1.3.2), we can can get

$$P(TN)\approx\prod_{i=1}^{n}P(TN_i|TN_{i-1}) \tag{1.3.3}$$

Where TN_i is a tag in the sequence of names (TN), for the likelihood probability can be estimated as :

$$P(SW|TN) \approx \prod_{i=1}^{n} P(SW_i|TN_i) \tag{1.3.4}$$

The above estimations was for a small sequence where TN_i is a tag in the sequence of names (TN) and SW_i is a tag at index i in a sequence words (SW). For the large training corpus, the needed step is estimate based on the number of times the tag occurs and the position of the tag in a given corpus.

$$P(T_i|T_{i-1}) = \frac{K(T_{i-1}, T_i)}{K(T_{i-1})} \tag{1.3.5}$$

Based on the training corpus, $K(T_{i-1}, T_i)$ is referred as a how many times the tag T_i occurs after the tag T_{i-1}. In the corpus, $K(T_{i-1})$ is considered as the number of occurrences for the tag T_{i-1}.

Therefore the estimation can be performed as follow:

$$P(C_i|T_i) = \frac{K(T_i, C_i)}{K(T_i)} \tag{1.3.6}$$

From the equation (1.3.6), the term $K(T_i, C_i)$ is referred as the sum of the times that a word "C_i" has a tag T_i in the training corpus. The process of computing the posterior using the above steps is called Markov model.

It is one of the most powerful statistical and machine learning (ML) techniques in modelling and high qualified in entities extraction. When the researcher is willing to train new data, HMM is very robust and efficient in computations. One of the limitations of HMM is that the researcher must have the notion of model topology and statistical techniques on how to deal with large amount of training data.

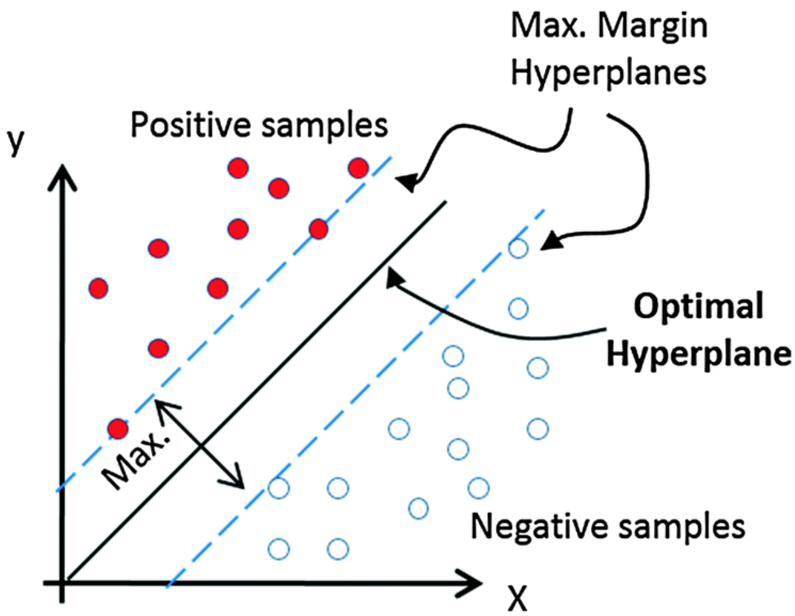
1.4 Supporting Vector Machine (SVM) based model

This model has an aim of classifying the named entities by separating the documents into two categories. The document must belong to one category, either positive or negative. SVM can classify linear data as well as non linear with a purpose of maximizing the margin between negative and positive documents.The plane which separate those two categories is called "hyperplane".

The main idea behind SVM modelling is to work with features and find the hyperplane. The hyperplane must separate all given samples regardless the dimensions.

1.4.1 Linear Supporting Vector Machine . For linear sample data, it is simple to plot the hyperplane to handle the separation. Data are spread separately between positive documents and negative documents. The way data are represented SVM decided whether to use linear modelling or not. IFRC reports are considered as multi dimensional documents.

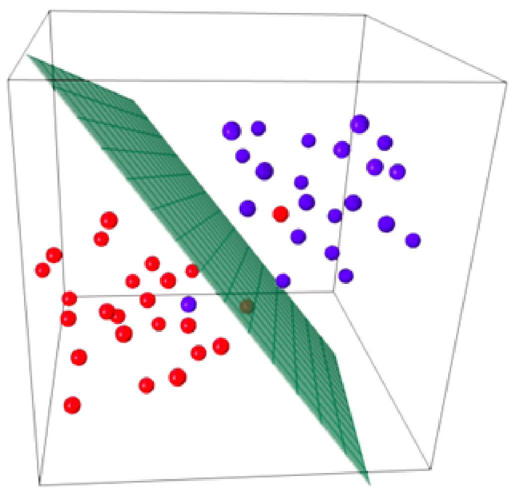
Figure 1.2: Two dimensional SVM (Moreira and Wichert, 2013)



In Figure 1.2, blue circles represent negative documents and red circles represent positive documents. The aim of SVM is to maximize the margin between negative documents and positive documents. Hyperplane is perpendicular. Optimal hyperplane separate perfectly two categories for two dimensional data.

Multidimensional documents whith separated classes are represented in Figure 1.3

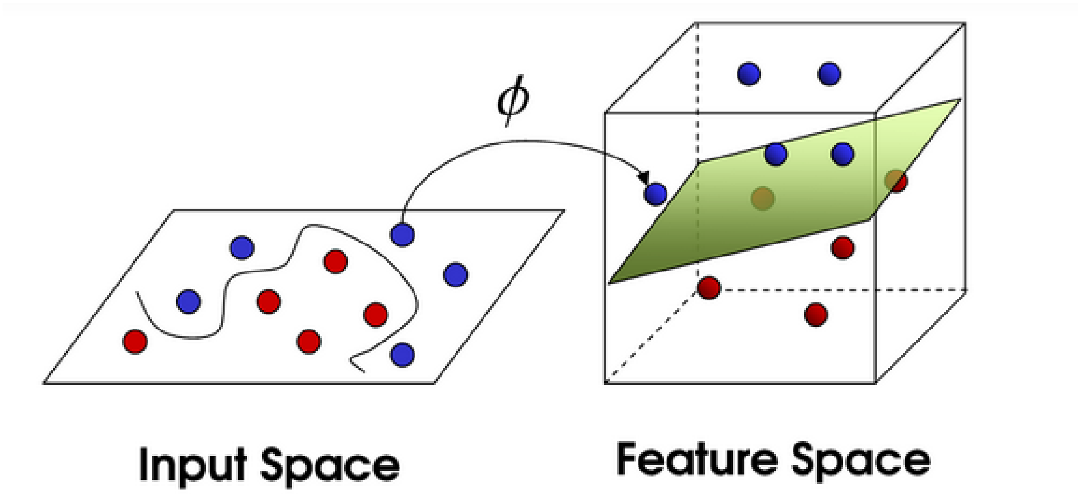
Figure 1.3: Multi dimensional SVM



Documents classes are clearly separated as shown in Figure 1.3. we have positive samples on right hand side and negative samples on left hand side. for multidimensional representation, hyperplane is a plane instead of a line. When documents are mixed, hyperplane can't not be neither a straight plane nor a straight line. samples documents are classified as non linear documents.

1.4.2 Non Linear data. Sometimes the representation of data is quite mixed way so that you cant plot hyperplane easily. When the hyperplane can not be plotted as a straight line, SVM looks for a way to linearise them by using a function. ϕ maps data to the higher dimensional space. Straightforwardly, the classification became linear. Figure 1.4 shows the way a function ϕ linearised the data.

Figure 1.4: Nonlinear SVM classification (Moreira and Wichert, 2013)



However, SVM has some disadvantages in classification. Some particular documents can not be easily performed without destroying the constructed weights but this can not happen for hand-written rule model. Machine learning uses decision tree procedure rather than SVM.

1.5 Extraction before Machine learning

Before the evolution of machine learning algorithms, NLP was using some techniques to extract needed information in documents like:

Hand-written rule

It is one of the standard approaches of NER and IE, it has been used for extracting the patterns from automated pages such as amazon, NLP was using hand- written regular expressions for unstructured humman-written text by delivering part-of-speech (POS), syntactic parsing and categories of semantic words.

Rule /pattern based extraction

Many IE systems uses rule/pattern to extract words and also phrases by looking to the context of those words or based on the their surroundings.(Califf and Mooney, 2003). Some system decided if the procedure of extracting the words should rely on the meaning of each word independently or on the context of their surroundings in a phrase. The limitation of this method is that some words do not have a closer mining to their surroundings that is why Patwardhan Siddharth with

Ellen Rilo in workshop called "ACL 2006" presented another approach which was generating an automated IE system to learn patterns from a large fixed data set within a specific domain (Patwardhan and Riloff, 2007)

Our research deals with reports generated through a template, compared to the work of (Patwardhan and Riloff, 2007) templates usages is a limitation.

1.6 Text classification and Naive Bayes

It is one of the most important algorithm in text classification by using base rule and bag of words to classify the entities (Manning, 2012).The user instead of going through the report and start posing many queries, text classification algorithm transient the need information. Its aims is to build a function θ which takes the bag of words and returns the class of sentiment C either positive or negative.

$$\theta$$

$$\Updownarrow$$

ARCS initiated its response immediately after the earthquake struck to address the immediate needs. The National Society (NS) regional branches were at the forefront of the response and worked with Disaster Response Units (DRU). ARCS staff and volunteers were deployed promptly to support rescue efforts, provide first aid to the injured and distribute immediate relief supplies to affected people alongside undertaking initial assessments. A total of 900 volunteers were mobilised to support this response operation. ARCS also supported to transport critically injured people to hospital and mobilized community members for voluntary non-remunerated blood donations.

$$\Updownarrow$$

$$\mathrm{C}$$

The classes of documents are mostly used in sentiment analysis. For example for social media where readers can say if they like the article or not.

For class analysis, we look for every word. but there exist another way to consider some subset of word and ignore other words. The procedure is to look for all words and retrieve those which form the subsets. Bag of words are formed after throwing away all words except the subsets.The use of the function θ is for attributing to each item of the bag of words a sentiment. The function θ assigns every word a class based on high probability.

1.7 Machine learning for Named Entities

The natural language processing is not enough to handle the sophistication and ubiquity of textual data. Deep learning using machine learning techniques has been introduced to solve this problems. The advantages of machine learning for Named Entities:

- Manual extraction of entities is too expensive.

- Fast processes of extraction.
- Extraction is done by learning algorithms and natural language tools.
- No limitation of languages due to polyglot package .

2. Research Methodology

2.1 Data and tools

World wide non governmental organizations publish some of their reports on their official websites. Web scrapping is one of the ways to extract data from website to the local machines. We downloaded the reports about appeals in pdf format from IFRC website. We used R-scripts for web scraping form our co-supervisor Xavier Vollenweider

We downloaded 1262 reports which have been submitted between 1^{st} January 2015 and 31^{st} December 2016. To differentiate the reports, each report has a report Id but different reports can refer to the same appeal Id. Appeal Id is a unique number given to a specific disease. As an international organization which works on the largest humanitarian activities in the world, IFRC reports we have talk about disasters and cash transfer program. Cash Transfer Program (CTP) describes the money used by IFRC to buy food, shelter, etc. For example the mon

2.2 Supervised vs Unsupervised Machine Learning

- **Supervised** is a machine learning part which deals with "labelled data", data are categorized and classified.

  We have a csv document which summarizes the appeals that we have. The shape of this document is 25 columns and 3997 rows. The "CTP" feature indicate if the appeal is classified as a Cash Transfer Program document or not. Among 3997 appeals, 404 are CTP.

- **Unsupervised** can be defined as a way machine learning processes "unlabelled data". the data are unstructured, uncategorised and unclassified. The reports we have are good example of unlabelled data.

  - Clustering is a technique for analysing data by identifying hidden groups in a data set. The hidden groups helps the machine to classify them data into small groups called "cluster" based on similarities or relationship found in data (Dy and Brodley, 2004).

2.3 NLP Corpus Preprocessing

Portable document format (PDF) has content which can not be extracted and manipulated easily. Our data have to be changed into another format in order to pull the information we need. We managed to transform 1260 reports. Our folder has 1260 transformed reports files which is considered as dataset. For analysing the data, we used Python programming language.

Our data reported on different areas of the World such as continents, countries and cities. For example "Europe IB23102015 23Oct2015.txt" covers European continent , "Afghanistan MDRAF003 05Nov2015.txt" and "Japan 0 16Apr2016.txt" reported on specific countries and "Port au Prince country cluster 0 04Oct2016.txt" reported on the most populous city of Haiti.

Corpus is a set of large data which are semi-structured. To extract entities from corpus is simpler than to deal with unstructured data. To get the corpus we filtered the data by using unicode of utf-8 and removing non-printable characters.

To get compatible data, we have to filter using the Unicode provides canonical and compatible equivalence.

Regular Expressions are for searching matched patterns in string. in Python, regular expression has operations and modules like "re.py" aND so on. they are used to manipulate characters in strings. regular expressions use a backslash (”\”) to indicate a special form without invoking the meaning of the special form. There are many regular expressions functions but some of what we used the most are :

- re.split(): this function split pattern and return the list of string.
- re.search(): it returns matching objects.
- the match object ".end()": in a search string, it returns the end position of the match.
- the match object: in a search string, it returns the start position of the match.

ASCII stands for American Standard Code for Information Interchange. it is uses numbers to represent text by using 128 characters. Computer uses ASCII exist within unicode for storing texts easily. All ASCII uses unicode.

ASCII characters are used to send and receive the e-mails, for text files and data conversions.

Figure 2.1: ASCII TABLE (Witte, 2002)

Hex	Dec	Char		Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL	null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH	Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX	Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX	End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT	End of transmission	0x24	36	$	0x44	68	D	0x64	100	d
0x05	5	ENQ	Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK	Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace	0x28	40	(	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal tab	0x29	41	)	0x49	73	I	0x69	105	i
0x0A	10	LF	New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2	Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3	Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4	Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB	End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN	Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM	End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB	Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC	Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS	File separator	0x3C	60	<	0x5C	92	\	0x7C	124	|
0x1D	29	GS	Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS	Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US	Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

Figure 2.1 demonstrates Unicode standard. It provides a unique number to each character composing a text regardless the language, program or platform.

UTF stands for Unicode Transformation Format. Due to the fact that unicode can't fit into one 8-bit bytes, there are many different types of UTF which store unicode in byte of sequence. characters are set into binary values 0 and 1. UTF-8 for encoding 8 bytes, UTF-16 for encoding 16 bytes and UTF-32 which is a standard for encoding 32-bytes are three current standards. For our corpus we used UTF-8.

After getting semi-structured documents, we removed the stop-words

which are defined as unnecessary words for extraction of entities from corpus.

Normally the stop-words return vast amount of unwanted information. Some example of English Stop Words: almost, are, or, details, during, upon and so on.

Now we can check how for all of 1260 documents and count the Stop Words to be removed from vocabularies of corpus. We trained corpus by nltk package called FreqDist which uses frequency distribution of each word occurs in corpus, then the module of nltk technique called "nltk.corpus.PlaintextCorpusReader" helped us to get total 58104 stop words over the whole 7796263 vocabularies.

2.4 Modules and packages

The procedure of extracting entities requires many linguistic packages. Machine learning algorithms check grammar rules, punctuation marks and syntaxes. Some documents can have special characters such as emoticons for emotions. To choose the packages to use, it is recommended to understand clearly how they work and the content type of your documents. This is a list of packages we used for extracting IFRC entities.

- **os** : This module is known as miscellaneous operating system interfaces. os represents the functionality of operating system with independent functions such as os.path.isfile(), os.path.exist, os.path.isdir, etc.

  Its functions are important for building platform-independent programs. The programs written using os module can execute in Windows and Linux regardless the machine operating system.

- **nltk**: Natural Language Toolkit is one of core packages for linguistic modelling . With various important built-in functions nltk is able to manipulate documents. The main idea behind nltk is to use $nltk\_corpus$ to collect all documents as one dataset, then split the documents into sentences using ne_chunk and remove the stop-words by importing $stopword$ from $ntk.corpus$, lastly apply machine learning algorithms to extract entities.

- **PyPDF2** is able to extract specific information from a pdf document based on the section they belong to. This package locates top section, title, author, etc. It has many functions such as splitting documents pages, merge document pages, encrypt and decrypt documents and so on. It can be compared to pdftk

- **pandas** is an open source with high performance structure within various build in functions. Dataframe design for presenting many data in organized way. Pandas is powerfull in data analysis, flexible, fast and manipulation took for any language. In our research, We used pandas for making the frames of our data.

- **codecs** module offers unicode string for encoding and decoding. codecs is used for handling errors and gives freedom to access internal registry. Codecs are not limited to text but mostly are for text encodings which is for encoding text to bytes. Additionally, there exits codecs for encoding text to text, some codecs can encode and decode at the same time.

- **defaultdictionary** has basic content of difference between verbs, nouns, adjectives, adverbs etc. Defaultdict in Python a dictionary with default value for missing key instead of key error.

- **Python String Strip** is a module which has methods to returns a string with trailing text removed. Two methods to strip text on both sides, rstrip for right hand side and lstrip for left hand side. Trailling text can be unwanted space, extension, punctuation marks, etc.

  To indicate the position of the character to be stripped we use left(l.strip()) which removes the character at the beginning of a string or right(r.strip()) to remove the character at the end of the string.

- **regular expressions** regex is a module which finds out the patterns between strings by setting rules for text. bytecodes compile those pattern rules and execute using matching rules. example methods for re are explained into Chapter 2.3

- **polyglot** is used to extract entities from many languages. It is multilanguage application supporter built as natular language pepeline.

- **Stanford** is one of most brilliant algorithms to extract entities from documents corpus. it has classifier models, jar files which are free downloads. Stanford has many packages to handle linguistic problems.

2.5 Extraction of Entities

To extract entities we used default dictionary built in collection package of nltk. Our dataset now is a folder containing 1260 corpus files, we used nltk chruncker to get sets of lines from our corpus. let have a look for our sample document the way lines are split.

Figure 2.2 shows the 45 first lines of the sample document. each each line is ended by $'\backslash n'$.

Figure 2.2: Set of sentences


```
['DREF operation n MDRAF003 Glide n EQ-2015-000147-AFG\n', 'Date of Issue: 26 May 2016 Date of disaster: 26 October
2015\n', 'Operation start date: 3 November 2015 Operation end date: 2 March 2016\n', 'Operation budget: CHF 465,684
Current expenditure: CHF 379,353\n', 'Number of people affected: 65,653\n', '1\n', 'Number of people assisted: 14,0
00 people (2,000 families)\n', 'Host National Society(ies) present (n of volunteers, staff, branches):\n', 'The Afg
han Red Crescent Society (ARCS) has at least 1,800 staff, 25,000 volunteers and 34 provincial branches and\n', 'sev
en regional offices nationwide. A total of 13 branches of ARCS are involved in the earthquake response, with some
\n', '700 volunteers mobilized to support activities\n', 'to the benefit of affected people.\n', 'N of National Soc
ieties involved in the operation:\n', 'The International Federation of Red Cross and Red Crescent Societies (IFRC)
 with the Movement partner actively\n', 'involved in supporting the ARCS response. IFRC and ARCS also maintained go
od coordination with other movement\n', 'partners, the International Committee of the Red Cross (ICRC), partners wi
th present in Afghanistan that include the\n', 'Canadian Red Cross Society, Danish Red Cross, Norwegian Red Cross,
 and Qatar Red Crescent Society. However,\n', 'Red Crescent Society of the Islamic Republic of Iran, Red Cross Soci
ety of China and Turkish Red Crescent Society\n', 'do not have offices in Afghanistan but have supported the earthq
uake response through bilateral arrangements with\n', 'ARCS.\n', 'N of other partner organizations involved in the
 operation:\n', 'Afghanistan National and provincial Disaster Management Authorities, Ministry of Rural Rehabilitat
ion and\n', 'Development (MRRD), UN agencies (WFP, UNICEF, WHO), International Organization for Migration (IO
M),\n', 'International Rescue Committee (IRC), People in Need (PIN), Care International and Oxfam.\n', 'Partners wh
o have contributed to the replenishment of this DREF include Canadian Red Cross Society/\n', 'Canadian Government
 (DFATD), DG ECHO, and Netherland Red Cross/ Netherlands Government (SEF). The\n', 'unspent balance of CHF 86,331 w
ill be returned to the DREF pot.\n', 'A. Situation analysis\n', 'Description of the disaster\n', 'Around 13:40 loca
l time (UTC +4:30) on 26 October 2015, a magnitude 7.5 earthquake struck Badakhshan province\n', 'in the north-east
region of Afghanistan. Badakhshan, Nangarhar, Baghlan and Kunar provinces were ranked the most\n', 'affected provin
ces. The Afghanistan National Disaster Management Authority (ANDMA) coordinated the initial\n', 'assessments in par
tnership with in-country humanitarian partners.\n', 'Assessment reports indicated 117 deaths, 544 people injured, 1
2,794 homes damaged and 7,384 houses destroyed.\n', 'In Badakhshan province alone, more than 51,000 people were aff
ected. The province also reported to have the most\n', 'extensive damages to properties. Kunar and Nangarhar provin
ces were recorded to have the highest number of\n', 'deaths and casualties as a result of the earthquake. Food and
 non-food items (NFIs), emergency shelter, and\n', 'psychosocial support services were identified to be among the i
mmediate needs. As the country moved into winter\n', 'season, winterization materials were being prioritized in the
response plan. Access to the affected population\n', '1\n', 'Afghanistan Earthquake, OCHA Situation Report No. 3 (a
s of 12 November 2015)\n', 'DREF Final Report\n', 'Afghanistan: Earthquake\n', '\x0cremained the most significant c
hallenge in delivering humanitarian assistance in a timely and effective manner. With\n', 'the support of the gover
nment, roads were cleared to pave way for humanitarian actors to reach the earthquake\n']
```


2.6 Top Section Dataset

From the analysis of IFRC pdf reports, most of them have a small table on the top. this table gives the image of what the report is talking about. This table summarizes what the document is talking about. For example the total amount of money spent in recovering a disease, the number of people who participated in a given activity, the location and so on.

While we were transforming the pdf data into txt format, this table occupied almost 25 first lines. Due to the limited time of the research, We decided to split those twenty five first lines of each document. the collection of those first twenty five documents has been considered as our new corpus.

Now we can use one of the algorithms to extract entities and for classification.

2.7 Stanford Named Entities Recognition

The data to be trained is unlabelled. Named Entities Recognizer labels the data to be extracted easily. it recognises sequence of words and its classification is mainly to name of persons, localization and organization.

Stanford Named Entities Recognition is an extractor implemented in java. It takes the sequence of words and label them Stanford named entities recognition is a able to identify correctly the named recogniser which labels sequences of words in a text. The next step is to split the sentences into set of words called tokens. By using the Stanford NER tokenizer where token can be tagged.

- **Stanford NER Tagger** is a package which has modules for classifying tokens with the taggs. A tagg can be defined as one of classes of significant words like nouns, adjectives etc. we used the package Stanford POS Tagger to classify the words.
- **Stanford NER Models** are many Stanford has different models such as "stanford-corenlp-full-2016-10-31", "stanford-ner-2014-01-04" which is the version we used.
- **Stanford Classifier** is a package which classify the entities into defined categories. It has four specific classes such as "Locations", "Persons", "Organizations" and "Others".

We specified the named entities that we wanted to extract. We classified them into the four categories by Stanford classifier. The last category called "others" combined all numerical entities such as time, amount of money, number of people, percentage, etc.

The reports from our corps are order by appeal numbers, the entities are in classified by Stanford algorithm.

Figure 2.3: IFRC entities from Stanford NER

	- Global - MAA00001 22Jul2015.txt	**- Global - MAA00006 24Apr2015.txt**	**- Global - MAA00010 10Nov2015.txt**	**- Global - MAA00021 02Jun2015.txt**	**- Global - MAA00028 01May2015.txt**	**- Global - MAA00029 21Jun2016.txt**	**- Global - MAA00040 02Jun2015.txt**	**- Global - MAA00040 10Nov2015.txt**
locations	[Neonatal]	[Geneva, Geneva]	[Bolivia]	[Sendai, Japan, Geneva, Cali, Colombia]	[Geneva, Panama, Kuala Lumpur, Nairobi, Dubai,...	[Syria, Iraq, Afghanistan, Libya, Ukraine, Yem...	NaN	NaN
organizations	[Global Health Report Health Department 2014 T...	[National Societies, NSKD, International Feder...	[DREF 2013 Number Amount, Red Cross Red Cresce...	[Preparatory Committee of WCDRR, DRR, Fourth G...	[IFRC Global Logistics Service, IFRC, National...	[Red Cross, Red Crescent, IFRC, IFRC, Middle E...	[Federation of Red Cross, Rules for Disaster R...	[IFRC, Crisis Management Department (DCM) Gl...
other	NaN	[2014, 2014, 2014 The Difference Overview The ...	[31 %, 69 %, 2014, 2 per cent, 2013, April 201...	[2015, March 2015, July, November 2014, June, ...	[2015, Strategy 2020, 2015, 2014, 2015, 2014, ...	[2014, 2014, 1990, 7 %, 2014]	NaN	[January June 2015, January 2015 12 months 72]
persons	NaN	NaN	NaN	NaN	[Sierra Leone]	[Jaime Sepulveda, Christopher Murray]	NaN	[Pankaj Mishra, Hakan Karay]

From Figure 2.3, Consider the for the report "-Global MAA00029 21Jun2016.txt", locations row shows that the report covered Syria, Irak, Afganistan, Libia, Ukraine, Yemen, etc. The extraction of entities separates clearly the categories.

2.8 Natural Language ToolKit (NLTK)

Natural Language ToolKit is one of the algorithm to extract named entities. It has different modules which are used to process the data alongside the extraction. NLTK chunkparser is a one of nltk module which uses Regular expressions. NLTK tokenize which splits the sentences into small units called tokens. This module helps the NLTK tagger to identify words independently.

Generally NLTK classify the entities into four categories which are known as Locations, Organizations, Persons and Others.

Figure 2.4: IFRC entities from NLTK

	- Global - MAA00001 22Jul2015.txt	**- Global - MAA00006 24Apr2015.txt**	**- Global - MAA00010 10Nov2015.txt**	**- Global - MAA00021 02Jun2015.txt**	**- Global - MAA00028 01May2015.txt**	**- Global - MAA00029 21Jun2016.txt**	**- Global - MAA00040 02Jun2015.txt**	**- Global - MAA00040 10Nov2015.txt**
locations	NaN	NaN	NaN	NaN	NaN	[West Africa, West Africa, Caribbean]	NaN	NaN
organizations	[Global Health Report Health, Contents, CBHFA,...	[Global, Difference, National Society, NSKD, N...	[Overview Statistics, DREF, CHF Total, DREF, D...	[DRR, HFA2, DRR, WCDRR, WCDRR, HFA2, UNISDR, W...	[IFRC Global Logistics Service, GLS, IFRC, Nat...	[oPt, Ebola Virus Disease, EVD, Sahel, Horn, R...	[Red Cross, Red Crescent, MAA00040, DCMs, DCM,...	[DEVELOPMENT, UPDATE, INTERVENTION, DCM, CHF, ...
other	[Disease, Maternal, Neonatal, Child, Sanitation]	[Geneva, Long, Geneva]	[Bolivia, Bolivian]	[Sendai, Japan, Geneva, Cali, Colombia]	[Geneva, Panama, Dubai, Las Palmas, Ebola, Ira...	[Iraq, Afghanistan, Libya, Palestinian, Yemen,...	NaN	NaN
persons	[Annual, Annexes Annex, Health, First Aid]	[Knowledge Development Division, Term Planning...	[Start, Red Cross Red Crescent, Emergency Fund...	[Billion Coalition, Climate, Climate]	[Overview, Kuala Lumpur, Nairobi, Guinea, Arab]	[Latin America, Global Health, Jaime Sepulveda...	[Crisis Management, Rules, Disaster Relief, Gl...	[Disaster, Crisis Management Department, Globa...

From Figure 2.4, Consider organizations extracted from the report "-Global-MAA00021 02 Jun 2015.txt", NLTK entities classifier was able to extract DRR, HFAR, WCDRR, HFAR2, UNISDR, etc. The classifier uses nltk tagger and default dictionary which help it to identify the names, verbs and adjectives.

For NLTK algorithms, some confusion between two categories "Location" and "Others" which might be fixed later.

2.9 Polyglot Named classifier

Compared to previous entities extractor, Polyglot has only three categories which are "Persons", "Locations" and "Organizations". For nltk, any entity which is not classified into those three categories is not considered as named entity.

Figure 2.5: IFRC entities from Polyglot

	- Global - MAA00001 22Jul2015.txt	- Global - MAA00006 24Apr2015.txt	- Global - MAA00010 10Nov2015.txt	- Global - MAA00021 02Jun2015.txt	- Global - MAA00028 01May2015.txt	- Global - MAA00029 21Jun2016.txt	- Global - MAA00040 02Jun2015.txt	- Global - MAA00040 10Nov2015.txt
locations	NaN	[Geneva, Geneva]	[Bolivia, Bolivia]	[Sendai, Japan, Geneva, Cali , Colombia, Cali]	[Geneva, Panama, Kuala Lumpur, Nairobi, Dubai,...	[Syria, Iraq, Afghanistan, Libya, Ukraine, Yem...	NaN	NaN
organizations	[Health, First, Adolescent, Sanitation, Cross,...	[Global, National Society and Knowledge Develo...	[Crescent, Cross, Red Crescent Societies, Disa...	[World Conference, UNISDR, WCDRR, Community Re...	[Logistics, Global Logistics Service, National...	[Global Health]	[nternational, of Red, Red Crescent, Managemen...	[Crisis Management Department, Crisis Management]
persons	NaN	NaN	NaN	[DRR]	[GLS, GLS, GLS]	[Jaime Sepulveda, Christopher Murray]	NaN	[Simon Eccleshall, Pankaj Mishra, Hakan Karay,...

Let us take an example report "-Global-MAA000029 21 Jun 2016.txt" from Figure 2.5, the entities which are classified as "Persons" Jaime, Sepulveda and Christoper Murray.

3. Results Discussion and Testing

3.1 General Overview

To extract and classify entities, We used Stanford, NLTK and Polyglot. These entities extractor have common categories which are "Persons", "Location" and "Organization", Additionally NLTK and Stanford NER has another category which is called "others". This last category is not very clear. It combines numbers, percentage and unclassified entities. This can cause the confusion for to the organization. The core categories are those three first groups.

Among these three entities extractor, Stanford requires time to run compared to others.

The named entities must be set by the organization based on its interest. Some reports are composed by many pages but some few point must be highlighted. Templates in reporting are important, they made life easy.

Before extracting the entities, You must know what the document is talking about. What the organization is struggling to know from the report.

Named entities from NLTK, Polyglot and Stanford are useful. They tried to summarise the primary information such as locations, persons and organizations.

Sometimes, extracted named entities are not sufficient. Regular expressions can be used to respond perfectly the will of the organization.

3.2 Case Study Results

After analysing 1260 documents, Let us take one sample file and work on top section composed by 25 lines.

Consider a document which is specific to African region. "Africa regional office MDR60002 03 Nov2015.txt". We are requested to extract name of Persons who participated in IFRC activities.

We had a function to extract four categories of entities by Stanford NER. It is only to specify the category we are interested in. To identify persons names manually is also possible.

3.3 Testing

For the security purpose testing gives a guarantee of correctness. It is a major chapter for assertion of the research quality.

The process of extracting entities can be done in different ways. Either manually or by the use of machine learning algorithms. The manual way has many disadvantages as explained in Chapter 1.7.

Computer algorithms have impact for solving human problems. However we have to do a comparison for a small dataset between algorithm results and human results. The correctness of a tested dataset gives a confidence for remaining datasets.

IFRC uses the templates formats to produce their report. It is way of structuring a content of the document. The use of templates made most IFRC reports to have almost the same size of top section. Top section contains important summary as explained in Chapter 2.6 .

Due to the time limitations, We tested some sample documents and We concluded for all top sections of the reports.

JSON file

By taking the sample file, We extracted names of entities in JSON format. JSON stands for Java Script Object Notation. It is built based on two universal data structures such as a pair composed by a name and a value, and ordered list of values which is considered as an array, sequence, vector or list.

Figure 3.1: JSON File Structure (Bray, 2014)

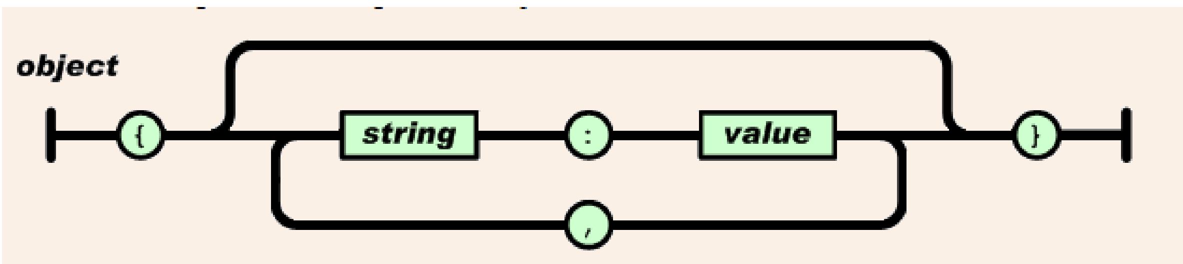


Figure 3.1 refers to the structure of our json file. It contains a small dictionary which has one feature of proper names.

In Machine Learning, there are three ways of testing the quality of algorithms. As We extracted entities from IFRC reports, to be sure on the work of algorithms, We calculated recall, precision and accuracy.

Precision has been calculated as a fraction of relevant instances over retrieved instances.

Recall has been gotten as a fraction of retrieved relevant instances over sum of relevant instances.

Prediction is made by algorithms to predict the name of persons in sample document. The correctness can be calculated based on comparison between what predicted and what extracted by hands.

We extracted manually three people who participate in IFRC sample report.

```
{'BothNames': {'0': 'Mamadou Basilah',
  '1': 'Tommy Trenchard',
  '2': 'Norbert Allale'}}
```

Figure 3.2: Comparison of names extracted by hands and Polyglot

After extracting three proper names as the Figure 3.2 shows, We are now going to do a comparison. We can compare the output of the algorithms.

Figure 3.3: Comparison between hands and Stanford NER

	Hand-labeled True BothNames	**Stanford NERC Authors**
0	Norbert Allale	Mamadou Basilah
1	Tommy Trenchard	Norbert Allale
2	Mamadou Basilah	Norbert Allale

Figure 3.4: Comparison of names extracted by hands and Polyglot

	Hand-labeled True BothNames	**Polyglot NERC Authors**
0	Norbert Allale	Mamadou Basilah
1	Tommy Trenchard	Tommy Trenchard
2	Mamadou Basilah	Norbert Allale

Figure 3.5: Test of Polyglot compared to hands extraction

```
The accuracy is 1.0
The recall is 1.0
The precision is 1.0
```

	Predicted Negative	Predicted Positive
Negative Cases	0	0.0
Positive Cases	0	3.0

Figure 3.6: Test of Stanford NER compared to hands extraction

```
The accuracy is 0.6666666666666666
The recall is 0.6666666666666666
The precision is 1.0
```

	Predicted Negative	Predicted Positive
Negative Cases	0	0.0
Positive Cases	1	2.0

Summarized graph of personal names extracted using three machine learning algorithms compared to hand extraction.

Figure 3.7: Combined

	Hand-labeled True BothNames	Stanford NERC Authors	Polyglot NERC Authors	NLTKStandard NERC Authors
0	Norbert Allale	Mamadou Basilah	Mamadou Basilah	Mamadou Basilah
1	Tommy Trenchard	Norbert Allale	Tommy Trenchard	Tommy Trenchard
2	Mamadou Basilah	Norbert Allale	Norbert Allale	Norbert Allale Point

4. Conclusion and Future work

Entities extraction has been performed using natural language toolkit, polyglot and Stanford named entity recognition. Evaluation of entity extraction is normally done by the metrics of precision, accuracy and recall between algorithms and named extracted by human hands. This research argues that top section of report has meaningful metrics. The results demonstrate that a process of extracting names of persons in top section of reports was well done.

As future work, the next step for entity extraction is to work on other sections of a document. To combine all used approaches into a software which can automatically visualised entity named by organization such as budget, number of people suffered from a disaster etc.

References

Shumeet Baluja, Vibhu O Mittal, and Rahul Sukthankar. Applying machine learning for high-performance named-entity extraction. *Computational Intelligence*, 16(4):586–595, 2000.

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

Tim Bray. The javascript object notation (json) data interchange format. 2014.

Mary Elaine Califf and Raymond J Mooney. Bottom-up relational learning of pattern matching rules for information extraction. *Journal of Machine Learning Research*, 4(Jun):177–210, 2003.

Jennifer G Dy and Carla E Brodley. Feature selection for unsupervised learning. *Journal of machine learning research*, 5(Aug):845–889, 2004.

Geoff Gordon. Support vector machines and kernel methods. *Online]. Dosegljiv: https://www. cs. cmu. edu/˜ ggordon/SVMs/new-svms-andkernels. pdf [Dostopano 28. 6. 2016]*, 2004.

Ralph Grishman and Beth Sundheim. Design of the muc-6 evaluation. In *Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996*, pages 413–422. Association for Computational Linguistics, 1996.

Christopher Manning. Information extraction and named entity recognition, 2012.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. Text classification and naive bayes. *Introduction to information retrieval*, 1:6, 2008.

Raymond J Mooney and Razvan Bunescu. Mining knowledge from text using information extraction. *ACM SIGKDD explorations newsletter*, 7(1):3–10, 2005.

Catarina Moreira and Andreas Wichert. Finding academic experts on a multisensor approach using shannon's entropy. *Expert Systems with Applications*, 40(14):5740–5754, 2013.

David D Palmer and Marti A Hearst. Adaptive sentence boundary disambiguation. In *Proceedings of the fourth conference on Applied natural language processing*, pages 78–83. Association for Computational Linguistics, 1994.

Siddharth Patwardhan and Ellen Riloff. Effective information extraction with semantic affinity patterns and relevant regions. In *EMNLP-CoNLL*, volume 7, pages 717–727, 2007.

Rahul Sharnagat. Named entity recognition: A literature survey. *Center For Indian Language Technology*, 2014.

Robert A Witte. *Electronic test instruments: analog and digital measurements*. Prentice Hall, 2002.