# POLITECNICO DI TORINO

Master of Science Program in Telematics Engineering

COMPUTER AND COMMUNICATION

NETWORKS ENGINEERING


MASTER THESIS

# HYBRID STRUCTURE BASED AND LEARNING BASED HATE SPEECH AND OFFENSIVE LANGUAGE CLASSIFICATION SYSTEM

NEW TECHNIQUE FOR NATURAL LANGUAGE FEATURE EXTRACTION (LSFES)

Candidate:

Waleed A. M. Alromaema

Supervisors:

Prof. Jussara M. Almeida

Prof. Claudio Ettore Casetti

Dr. Luca Vassio

Academic Year 2017/2018

# ABSTRACT

This thesis research proposes a new methodology for Hate Speech and offensive language classification system by development of a hybrid system composed of Structured Based Analysis for development of an OIE (Open Information Extraction) system for linguistic feature extraction and Learning Based Classification System.

The underlying idea is to develop a structured based system for feature extraction using the grammatical representations of the sentences by deep analysis of the hierarchal structure of Dependency Parse Tree and applying some heuristic algorithms using some rules to extract the hidden meaning within the textual formation and extracting some predefined templates structure that have been selected based on manual analysis of hate speech dataset using Consistency Parse Tree to identify the significant sentimental parts and phrases within the sentences.

Those features are prepared and feed into the classification based system for learning and predictions.

In machine learning, the features selection aims at effectively reduce the dimension of feature space in which each sample feature represents a vector in n-dimensional space, and During feature extraction, uncorrelated or superfluous features will be deleted. Text feature extraction plays a crucial role in text classification, directly influencing the accuracy of text classification and can better improve its accuracy.

So, our work path was aimed to use a new technique for feature preparation based on Semantics Analysis and Linguistics, which deals with the study of meaning in language, the relationships between words, phrases, and symbols, their indication and representation of the knowledge they signify.

In this regards we analyzed 1700 tweets to identify the positioning of the fundamental sentimental, hateful and offensive expression parts within the sentences that curry and contribute more in formation of the general meaning of sentence. This analysis study was aims to identify the linguistics template structure that contribute in concluding the overlay of the tweet sentimental orientation of being hate, offensive or neither, on this analysis we have parsed more than 1700 tweet and generate the consistency parse tree browsing the hierarchal structure of each tweet annotated by POS (Part of Speech), Constituency grammars principle is that a sentence can be represented by several constituents derived from it. These grammars can be used to model or represent the internal structure of sentences in terms of a hierarchically ordered structure of their constituents. every word usually belongs to a specific lexical category in the case and forms the head word of different phrases. The result of this analysis is a set of structured templates that currying the significant parts of sentences.

We have developed an OIE (Open Information Extraction) for feature extraction by development of some heuristics algorithms for templates extraction by establishing and formulating a set of assumptions and manipulation rules on top of the dependency parse tree, Dependency Parse Tree is built based on the dependency grammars, these grammars do not focus on constituents like words, phrases, and clauses but place more emphasis on words. The result of the heuristics algorithms is the pairs, triples of the recognized templates that can be represented as a feature vector to the classifiers.

The advantage of this new technique is that the feature selected have a direct relation to the semantics of sentences so that it can be used to weight the terms as function of its importance and its terms semantical relations. We have used features generated by LSFES together with other related features that relay on semantic meaning of the sentences as n-gram, TFIDF, sentimental features, hate score level, POS, semantic

features of user behavior and word2vector embedding as a features applied to a set of machine learning classifiers It shows a good classification results.

# ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my supervisor Professor. Claudio Casitti, Associate Professor, Telecommunication Engineering Department, Polytechnic di Torino and Supervisor Professor. Jussara M. Almedia, Associated Professor, Computer science department, Universidade Federal de Minas Gerais, for their support throughout my Master thesis, for their patience, motivation, and immense knowledge. their guidance helped me through all the time of the research and the writing process of this thesis. I could not have imagined having a better advisors and mentor for my master thesis like them. Similar great gratitude goes to the assistance PHD student Dr. Luca Vasio for his willingness for help and support.

To all professors that teach me during my master study. To all the researchers that make their work freely available to others. I pass all my success to my father's soul, and deeply appreciate my mother, grandfather and grandmother sacrifices that they have made on my behalf. I would like to express tremendous appreciation to all of my family for their continuous and unfailing prayers and support. To my little girl and my little son for their miss being fare from them along the master course. Thanks to all the friends and colleagues that always provided me happiness. To Edisu for their support, grant and offering study environment for all students. To Polytechnic secretary for assistance and supports.

Finally, a great and extreme appreciation to my homeland country 'Yemen' its support throughout master study. That's was what sustained me this far. We pray all the time to rescue and protect you from this political unfair ignorant war.

Waleed Alromaema.

# TABLE OF CONTENTS

# III. THIRD PART (LEARNING BASED SYSTEM)

# IV. FOURTH PART

## (HYBIRD STRUCTURE BASED AND LEARNING BASED HATE SPEECH AND OFFENSIVE LANGUAGE CLASSIFICATION SYSTEM)

# LIST OF FIGURES

# LIST OF TABLES

# I. PART 1: (INTRODUCTION).

In this part we presented the common parts of the thesis in general. we have introduced an introduction to hate speech and Information extraction we also formulate some thesis research Assumptions and Questions and presented our thesis objectives to be achieved.

we also have presented related work in hate speech and offensive language and show how IE (Information Extraction) compatible and contribute to improve learning performance.

# CHAPTER 1.

# (INTRODUCTIONS).

## 1.1. Introduction to Hate Speech.

with the increase of internet usage and social media platform, the worldwide statistics on global digital report on 2018 stated from the total population of 7.593 billion there are 4.021 billion internet users, 3.196 billion social media users, 5.135 billion mobile phone users, 2.953 billion active mobile phone user this present the worldwide connectivity become a dominant and internet accessibility become easier with increase of telecom services and internet service providers, the accessibility on Northern, Western and South Europe and North America have the largest internet penetration with between 74%-94% internet users compared to total population.

With the massive increase in social interactions on online social networks, there has also been an increase of hateful activities that exploit such infrastructure.

The study of hate speech in social media is particularly important specifically in middle east to study the extent of its impact on the revolutions and creating the creative chaos, which led to the destruction of cities, landmarks and expand patching war from political war to religions, ethnics and racism war. On my city, when the war has increase its activities, people have divide in parts based on their political and religious parties. Politics parts and religions parts utilizes each other to spreads its control over communities. An online hate speech on social media take place with increasing the revolutions. Poster and replier sent threats each other and online hate speech leads to street wars and action based hate speech (verbal form) leads to criminal activities, once haters equipped with weapons the action form of hate started and ends in a war that results on hundreds of social media users, followers, posters was died including their families, women's and children's randomly without distinctions or exclusions. Social media platforms accelerate social war by providing a fast communication environment, creating events sharing hatred posts, frightening and scaring images and videos that attracted peoples and communities' emotions, feeling and their affiliation to politics, ethics origin, racist and religious parties. These online activities affected people Psychologically and behavioral to currying a negative and hatred energy and have moved those scenarios to the real life criminal activities. It was beneficial for social media platforms to add an application that quote and stop or prevent such scenarios by add-ins applications.

It is interesting to study hate speech and offensive language specifically on those regions that don't respect hate speech national and international laws and social media terms of use to limit hater and criminal activities.

Hate speech and offensive classification is a crucial task where hate speech is an offensive terms used in different context that what makes it difficult for classification.

In this context utilizing IE (Information Extraction) for attempting to extract the context by syntactic and semantics analysis on linguistic of sentence could help in machine learning classification performance.

So, we can see and track how our emotions, feeling, positive and negative speech can be diffused and distributed on the social networks, how negative hate speech can be distributed faster than positive, societies being aware of hate speech and such posts got more attractions than positive one , human statuses, feeling and emotions can change over time based on recent activities followers posts, news feeds and reads, and twitterer role in community have its influence based on their authorities e.g. Tramp tweets posts vs Kim Gone can lead to third world war.

So social media analysis takes its importance for researchers to study its social structure made up of a set of social actors as individuals, groups or organizations, sets of dyadic ties, and other social interactions between actors. Social networks and the analysis of them is an inherently interdisciplinary academic field which emerged from social psychology, sociology, statistics, and graph theory. Social network analysis is now one of the major paradigms in contemporary sociology, and is also employed in a number of other social and formal sciences.

This study gives us a perspective view of the social relation and hate orientations. The goal of hate speech language study in social media is to be used for design a preventive system or alerting systems about the terms of usage on social media platforms.

## 1.2. Introduction to Information Extraction.

Information extraction (IE) is the task of automatically extracting structured information from unstructured or semi-structured data. IE is the task of processing human language texts by means of natural language processing (NLP). Information Extraction, is the process of attempting to make the computer to understand Human Natural Language information.

It is interesting to note that most of social media and microblogging platforms state in its public post user interface “What’s in your mind” and users before they write down or express their thoughts start forming ideas and facts in their head and attempting to structure and express them in textual form of specific language. Poster (users) organize facts and relationships between them to form ideas in logical form by following the syntax and semantics of the written language and the reverse process of the readers that attempt to decode this information into facts or ideas to be understood. This what Information extraction aims to do, the role of writer (in terms of Natural Language Generation) and reader (in terms of Natural Language Understanding). This can be done by extensive knowledge of linguistics and natural language construction rules and grammars, phenomenon and related task of NL understanding that to be a complex task as Human language is the channel to transform his thoughts, feeling, emotions, ideas and logic that is completely difficult to be analyzed and or understood since it builds based on random real life events and human knowledge base and old memories.

Information Extraction attempt to find how one could use a computer for tackling natural language related tasks by providing an advanced tools and techniques to build an OIE (Open Information Extraction) for Natural Language analysis and understanding. For this reason, we attempted in our thesis to develop a tool that facilitate machine understanding of textual data to transform social media posts, tweets to be understood by machine learning system as a modeling to the reader task in previous poster and reader example in form of natural language feature extraction to machine learning system.

## 1.3. Research Assumptions.

Assumption 1. Hate Speech and offensive language classification is a Natural Language Processing and Machine Learning problem in witch classification accuracy and performance depend on ground truth of the labeled dataset at first and on fidelity of feature extraction from each record in labeled dataset.

Assumption 2. Features has a direct influence on NLP and ML systems performance. an existing features as sentimental, semantic, pattern based, n-gram, meta features and user behavior features are all just a trial to link between natural text meaning and features values, although features as syntax, pattern or even metadata don't have a direct link to text semantics.

Assumption 3. The direct methods in NLP of Feature Extraction that directly links text with feature are BOW (Bag Of Word), N-gram method and TF-IDE (Term frequency inverse document frequency). Those methods generate feature vocabulary from text that contains all adjacent pairs, triples and n-gram without regards to its contents or meaning so that feature vector will contain for relevant and none relevant feature at most.

Assumption 4. Utilizing a new technique that directly maps text meaning into meaningful feature vector have a direct influence on machine learning accuracy and learning efficiency. This involves development of an OIE system for Natural Language Linguistics Analysis and Extraction.

Assumption 5. The compatibility between different stages of machines learning pipeline and the dataflow engineering of the interface between different stages is significantly important, starting by dataset collection, high level feature selection, feature preparation, vectorization, dimensionality reduction, classifiers selection and fitting classifiers.

Assumption 6. Mathematical behaviors of different machine learning classifiers and its input features representation have an impact on classification output performance. So the expertise of witch classification algorithm work best for a particular application depend on the engineering behind features values input and classifiers mathematical processing functionality.

Assumption 7. In feature extraction and preparation, the process of identifying precisely the direct relevant terms and the relationship between them eliminating non relevant data from our focus have an impact on the system accuracy and that as much as we understand the meaning of the terms and its relation forming these relations in a way that understood by the underlain machine learning system the detection accuracy will converges to 100%.

## 1.4. Research Questions.

Q1. Hate Speech and Offensive Language seems to be the same class with the same terms used almost in both but in different context. It was difficult also for experts to distinguish between the two classes during annotation and labeling of the datasets. How can we define a new technique that can extract the semantics of the speech and the contexts in witch terms are represented on to facilitate classification task?

Q2. What are features that we expect to improve the classification accuracy of the system and can reduce the dimensionality of the feature space?

Q3. Text feature extraction plays a crucial role in text classification, directly influencing the accuracy of text classification. It is based on VSM (vector space model), in which a text is viewed as a dot in N-dimensional space. Datum of each dimension of the dot represents one (digitized) feature of the text. And the text features usually use a keyword set. It means that on the basis of a group of predefined keywords, we compute weights of the words in the text by certain methods and then form a digital vector, which is the feature vector of the text. Existing text feature extraction methods include filtration, fusion, mapping, and clustering method.

The question is that given the underline structure of feature representation using a (weighted digitized vector form) how can we increase the weight of the words as a function of its importance? how machine can understand the terms relevance and its meaning?

In this context deep analysis of Natural Language and Linguistics are needed. Development of new feature technique is a lengthy process and mapping between the semantic feature and digitized

representation is significant trick? How we can do this mapping of feature to the corresponding digitized weights vector?

Q4. The common vectorization process use range of n-gram to represent the ordered structure of the language to build its own vocabulary, are there any way to develop a custom vectorizer to add our own vocabulary generated by Linguistic structured based Analysis?

Q5. in machine learning there is an expertise on the selection of the classifiers that can match our feature sets for prediction, witch classifiers that expected to work well and why?

Q6. Questions above concentrated on development of a new technique for features extrication that relayed on the sentence semantic extraction and generation of a new meaningful constituents arranged in pairs or triples, can we develop a new OIE (Open Information Extraction) system for linguistic Feature Extraction?

## 1.5. Research Objectives.

- The main objective is to define a new methodology for hate speech and offensive language classification model by development of a hybrid system that composed by Natural Language Linguistics Structured Feature Extraction system and Learning Based System.
- Development of new novel OIE (Open Information Extraction) technique for Natural Language Linguistics Features Extraction.

## 1.6. Thesis outline.

The thesis consists of 4 Parts organized in 15 chapters. The first Part is an Introduction to Hate speech system and Information Extraction IE and contains for two chapters [ch1-2]. The second Part is (Structured Based System) and consists of 6 chapters [ch3-8]. The third Part intended for (Learning Based System) and consists of 5 chapters [ch9-13]. The fourth part is for the (Hybrid Structure Based and Learning Based Hate Speech and Offensive Language Classification System) it consists for two chapters [ch14-15].

- First part:
  - Chapter 1: we introduce an introduction to hate speech and Information extraction. we also formulate some research Assumptions and Questions that have been tackled in our thesis. We have also set thesis objectives that to be achieved by the end of thesis research work.
  - Chapter 2: we have presented related work in hate speech and offensive language and how IE (Information Extraction) can be utilized and contribute to improve learning performance.
- Second Part (Structured Based System):
  - Chapter 3: we present some principles of Language and its structure that includes an overview of language components and grammatical theories as sentence structure, consistency and dependency grammar that build the base of parsers development and manipulation rules.
  - Chapter 4: based on theoretical presentation we made in chapter 2 of language syntax and structure this chapter present a set of parsers used for language parsing .it is a lexical analysis and hypothesis that groups a set of language manipulation tasks in to a convention parser names. It is a base of some tools we will use in the following chapters as Stanford CoreNlp toolkit.
  - Chapter 5: This chapter is an earlier stages analysis of Hate speech dataset, by making syntactical analysis of tweets dataset and manually presenting consistency pars tree of 1700 tweet to identify the common template structure that currying the significant hate expressions to be used as baseline for our development of Heuristics algorithms of Linguistic Structure Feature Extraction system (LSFES) that we have developed as we will see in the subsequent chapters.

- Chapter 6: It is the first chapter into system design, this chapter is the core of structure based system design and its heuristics algorithms development. In this chapter we have prepared a set of assumptions on sentence parts organization and define a set of rules used for dependency parse tree manipulation for extraction of templates structure we have defined in analysis stage on chapter 5. we also described the heuristics algorithms used and its flowcharts.
- Chapter 7: this chapter describe the final system design the so called (Linguistic Structure Features Extraction System (LSFES)) as an integration of all Heuristics algorithms extraction.
- Chapter 8: in this chapter we have discussed and analyzed the LSFES system results by provisioning a case study analysis example and we have showed how sentence components connected each other in relations, how semantic network represented and how our system extracts this relation and prepare the features we have also motivated our feature extraction by philosophical example for improves learning performance. We have compared LSFES with an OIE systems as OpenIE and ClausIE in terms of functionality and use. We also discussed about future use of LSFES system including vectorizers, vocabulary extension, n-gram meaningful set generation and feature extraction.

❖ Third Part (Learning Based System):
- Chapter 9: in this chapter we described Machine Learning Algorithms used in our system and its mathematical behaviors, we think that this mathematical view could give an impression about witch algorithms suitable for a particular application.
- Chapter 10: we introduce an overview of hate speech and dataset collection we used.
- Chapter 11: in this chapter we investigate and analyzed feature extraction we used and motivate our selection of such features that related to semantic meaning and tightly coupled with hate expression. including features generated by our proposed system (LSFES Features).
- Chapter 12: Dimensionality reduction is the first step in machine learning pipeline and the preprocessing step into classifiers, we described DR methods related to Natural Language, it gives us and engineering view to the feature space representation and the methodology of removing irrelevant and redundant data. We take dimensionality reduction and feature space representation into account during our design engineering of feature interface generated by our system (LSFES) to increase learning accuracy.
- Chapter 13: Her, we have presented the Learning based system workflow and proposed approaches we used.

❖ Fourth Part (Hybrid Structure Based and Learning Based Hate Speech and Offensive Language Classification System):
- Chapter 14: we have presented the final hybrid system Methodology and the process of Integrating two systems we described on the two parts (Structure based system and learning based system) into one hybrid system and the required tools.
- Chapter 15: we have investigated and discussed the final system classification results and the performance measurement.

# CHAPTER 2.

# (RELATED WORK).

In machine learning and pattern recognition, a feature is an individual measurable property of a phenomenon being observed. Choosing informative, discriminating and independent features is a crucial step for effective algorithms in classification and regression. So our thesis study focus on features selection and generation for hate speech and offensive classification.

Working on machine learning classification model with different approaches relays on which features have been used so fare together with classification method that mainly focus on supervised learning.

In Natural Language Processing system, the most logical information feature to utilize are the high-level surface features at the level of language contents as Bag of words (BOW), n-grams feature and character n-gram features. These features reported to be highly predictive features and used by a majority of authors [1][2][3][4][5][6][7][8] (Waseem and Hovy, 2016; Burnap and Williams, 2016; Nobata et al., 2016 ,Van Hee et al., 2015; Hosseinmardi et al., 2015; Xu et al.,2012; Warner and Hirschberg, 2012; Sood et al., 2012b).

N-grams is an enhancements of BOW that allows to improve classifiers' performance, it incorporates at some degree the context of each word by generating a combination sequences of words into lists with size N, [1][2][3]( Waseem and Hovy. 2016; Pete Burnap and Matthew L. Williams.2016; Nobata 2016).

N-gram features can be further improved using character n-gram features to tackle the problem of misspelling variation words generated by users, Mehdad and Tetreault (2016) [10] find that character n-grams prove to be more predictive than n-grams., Nobata et al. (2016) [3] report that character n-gram and n-gram are further improving performance by combine it with other features.

Although that bag-of-words features yield a good classification performance, it lakes dynamics of online hate speech in which words in the bag induce the context and that Hate Speech discourse is not limited to the presence or absence of a fixed set of words, but is instead related to the context in which it appears. word generalization can be used in this context so that by using a clustering method assigning words to particular cluster Warner and Hirschberg (2012) [7], other work (Blei et al., 2003) [11] assign for each word a topic distribution and the degree of corresponding using LDA (Latent Dirichlet Allocation).
We need a model that exploit similarity between words under different uses and context, by using Neural Embedding, words represented by a vector in n-dimensional space so that similar semantically words having similar vectors (Mikolov et al., 2013) [12]. to get the feature of the entire tweet to exploit the context is by averaging the vectors of all words occurring in one tweet (Nobata et al., 2016) [3]. This vector replaces individual features vectors indicating the presence or frequency of particular words.

Other works use sentiment-based features where Hate Speech and negative sentiment closely related, several approaches as Dinakar et al. (2012) [15] and Sood et al. (2012b) [8] relates this relation by incorporating sentiment as a feature into hate speech classification as the first step Gitari et al. (2015)[13].

follow a. Further, (Thomas Davidson, Dana Warmsley 2017) [40] use positive, negative, and neutral sentiment values as feature values. (Sood et al., 2012b [8]; Burnap et al., 2013 [2]) show that hate speech presents a high degree of negative polarity.

Hate speech has a popular negative words and phrases that is frequently used as a common hate expression, there are many lexical resources publicly available in [noswearing, rsdb and hatebase organizations online] this lexical resources have been used by some authors as (Thomas Davidson, Dana Warmsley,2017 [40]; Burnap and Williams, 2015[2]; Xiang et al., 2012[6]; Nobata et al., 2016 [3]) as a features values represents scores level of use frequency.

It will be beneficial if the research of hate speech or any Natural Language based applications rely on analysis of the textual content finding linguistics features (syntactic and semantics) of the passage or tweets to release an advanced solution to the above features as BOW, n-gram, deep learning and word embedding, lexical resources corpus and many others. So many researchers (Edel Greevy and Alan F. Smeaton 2004 [14]; Thomas Davidson,2017 [40]; Karthik Dinakar, Roi Reichert,2011 [15]) have utilized a syntactical feature as Part-of-speech (POS) approaches that reflects the context and detect category of the word in the context of a sentence. These approaches detect frequent of POS use in bi-gram and n-gram in hate speech detection. Although that POS proved to not working properly in the classes identification (Peter Burnap and Matthew L. Williams. [2]).

Utilizing a deeper Linguistic feature benefit that extraction of non-consecutive words relations that potentially related a long-distances, Chen et al. (2012)[19]employ typed dependency relationships. for example (he is lower class bigs) dependency relation nsubj(pigs, he) links between a distanced target ‘he’ and the offensive term ‘pigs’. By knowledge that words of the sentence are syntactically related increase classifiers learning from the real meaning of sentences and more convey hate speech than using hate speech key words occurring in a sentence without any syntactic relation. Dependency relationships are utilized in the feature set by many authors (Gitari et al. (2015)[13], Burnap and Williams (2015-2016)[2] and Nobata et al. (2016)[3]).

Some work manually extract the relation by setting one argument as hate or offensive term and elaborate the second argument as in Chen et al. (2012) [19] and Gitari et al.(2015)[13] work.

Other work utilized a sentence structure template for pattern matching search of hate speech (Mainack, Leandro Araújo, Fabrício Benevenuto.2017) [20] they used some templates as I < intensity > < userintent >< hatetarget > that extract the common hate expressions in online posts this approach give high precision but low recall as a result of misses hate speech which does not conform to the defined sentence structure.

As a result, in our thesis research we aimed to develop a new OIE (Open Information Extraction) tool for Linguistic structure features extraction such that generated features generalized to match all possible hate template expressions, and preparation of a meaningful features set by analysis of all semantic relations and dependencies of the words in sentence. Such tool will cover to combine the advantages of most features extraction methods presented above and solves its common problems. Although Extracting or selecting features is a combination of art and science, feature engineering is involved to developing a Linguistic Feature Extraction system. It requires the experimentation of multiple possibilities and the combination of automated techniques with the intuition and knowledge of the Natural Language and Linguistics science. Automating this process is feature learning, where a machine not only uses features for learning, but learns the features itself. Such tool will be improved to be used as a Linguistic Feature Extraction in Natural Language Processing in general although we made its analysis on hate speech domain.

# II. SECOND PART: (LINGUSTIC STRUCTURE BASD SYSTEM ANALYSIS)

This part is a Project framework for establishment, analysis, design and implementation of an OIE (Open Information Extraction) tool for Linguistic Feature Extraction that to be used for Natural Language Feature Extraction. In the first three chapters [3-5] we presented some principles, phenomenon, rules assumptions on linguistic field as an analysis stage to be used as a base in the following chapters [6-8] of design and implementation. This developed framework the so called LSFES (Linguistic Structure Features Extraction System) can be used in feature for Natural Language Feature Extraction. Although, its analysis has been done in the field of hate speech. The tool has been utilized for hate speech dataset and the output used as a new feature input to the Learning Based System.

# CHAPTER 3.

# (LANGUAGE SYNTAX AND STRUCTURE ANALYSIS).

Usually the principals, conventions and specific rules are governed the way words combined together into phrases, phrases get combined into clauses and sentences, this structure is the basic unite of many areas like text processing, annotation, and parsing. In English, words usually combine together by syntactic rule like grammar to construct the constituents. Tweets texts or messages are all consists of a set of constituents that came together and are related to each other in a hierarchical structure. The meaning conveyed based on the order or position of the words. Parsers are used to construct those constituents using hierarchical sentence structure. In this section, we will describe the constituents building block components.

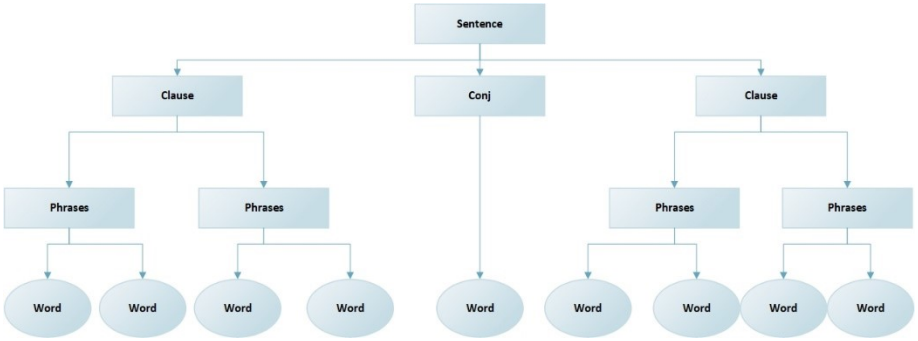



**Figure 1 Hierarchical Sentence Syntax Structure**

## 3.1. Words

In linguistics, a word is the smallest element that may be uttered in isolation with semantic, words can comprise morphemes witch are the smallest meaningful unit. It is useful to annotate and tag words and analyze them into their parts of speech (POS) to see the major syntactic categories. Here, we will cover the main categories and various POS tags. words can fall into one of the following major categories:

| Words Categories | Description |
|---|---|
| ***N (Noun)*** | Nouns are words relates some object or entity which are sensible or insensible. Linguistically, a noun is a member POS whose members can occur as the main word in the subject of a clause, the object of a verb, or the object of a preposition. [23] Furthermore, each POS tag like the noun (N) can be further subdivided into categories like singular nouns (NN), singular proper nouns (NNP), and plural nouns (NNS). The POS Tag of the Noun is N. |
| ***V (Verb)*** | Verbs are words that are used to describe certain actions, states changes, or occurrences. There are a wide variety of further subcategories, such as auxiliary, reflexive, and transitive verbs and others. The POS tag symbol for Verbs are *V.* |
| ***Adj (Adjective)*** | Adjectives are words used to describe or qualify other words, typically nouns and noun phrases. [24]. The POS tag symbol for adjectives is *ADJ or JJ.* |

| | |
|---|---|
| ***Adv (Adverb)*** | Adverbs act as modifiers for other words including nouns, adjectives, verbs, or other adverbs. The POS tag symbol for adverbs is *ADV.* |
| **CONJ (*Conjunction)*** | used to bind clauses to form sentences POS tag is CC. |
| **DET (*Determiner)*** | used to determine articles like *a, an, the,* and so on POS tag can be DET . |
| **PRON (*Pronoun)*** | words that represent or replace a noun POS tag PRP. |

*Table 1 Words Categories*

The basic tags N, V, ADJ and ADV are typical open classes belonging to an open vocabulary. *Open classes* are word classes that can be extended by accepting the addition of new words to the vocabulary which are invented by people through processes like *morphological derivation,* invention based on usage, and creating *compound lexemes.*

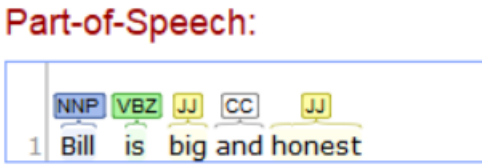



**Figure 2 Words with their POS annotation**

## 3.2. Phrases

In linguistic analysis, a phrase is a group of words (or possibly a single word) that functions as a constituent in the syntax of a sentence, a single unit within a grammatical hierarchy. Ordering the words in a form that give a meaning constructing a phrasal category, phrase represented by a main word called the Head. In the hierarchy tree, groups of words make up *phrases,* which form the third level in the syntax tree shown in figure [1]. There are five major categories of phrases as in figure [3]:

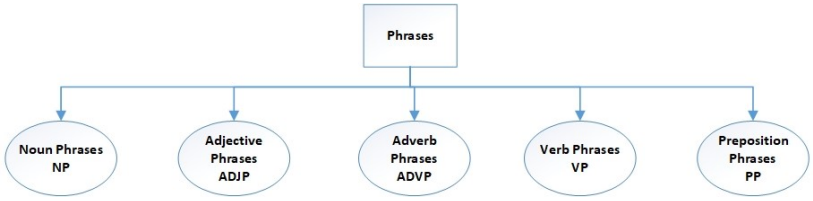



*Figure 3 Phrases Categories*

| **Phrases** | **Description** |
|---|---|
| ***Noun phrase (NP)*** | The Noun acts as the head word. Noun phrases act as a subject or object of the verb or can be replaced by a pronoun.[21] |
| ***Verb phrase (VP)*** | In linguistics, a verb phrase (VP) is a syntactic unit composed of at least one verb and its dependents objects, complements and other modifiers but not always including the subject. There are two forms of verb phrases. One has the verb components as well as other entities such as nouns, adjectives, or adverbs as parts of the object. The verb here is known as a finite verb. It acts as a single unit in the hierarchy tree and can function as the root in a clause. This form is prominent in constituency grammars. |

| | |
|---|---|
| | The other form is where the finite verb acts as the root of the entire clause and is prominent in dependency grammars. |
| ***Adjective phrase (ADJP)*** | Adjectives and adjective phrases function in two forms in clauses, either attributively or predicatively. Attributive, they appear inside a noun phrase and modify that noun phrase and used to describe or qualify nouns and pronouns in a sentence, and when they are predicative, they appear outside the noun phrase that they modify and typically follow a linking verb (copula). |
| ***Adverb phrase (ADVP)*** | In linguistics, an adverbial phrase is a group of two or more words operating adverbially, meaning that their syntactic function is to modify a verb, an adjective, or an adverb. Adverb acts as the head word in the phrase. |
| ***Prepositional phrase (PP)*** | phrases contain a preposition as the head word and other lexical components like nouns, pronouns, and so on. It acts like an adjective or adverb describing other words or phrases. |

*Table 2 Phrases Categories*

We will see how these five major syntactic categories of phrases used as building block of the language structure of dependency and consistency grammars that we will use to develop the heuristics algorithms of Hate Speech template structure Extraction. different kind of parsers can be used to extract these constituents.

These five phrases categories can be generated by applying several rules as a function of words, rules as grammars can be used.

## 3.3. Clauses

A *clause* is a set of words grouped together with some relation it contains a subject and a predicate in some cases the subject cannot be present, and the predicate has a verb phrase or a verb with an object. one or more clauses can be combined to form a sentence.

Clauses can be classified into two distinct categories, *main clause* and *subordinate clause. The main clause is an independent clause that works as sentence,* the subordinate is a *dependent* clause that depend on other clauses and joined with subordinating conjunctions. clauses can be subdivided into several categories based on syntax:

| Clause | Description |
|---|---|
| ***Declarative*** | These clauses do not have any specific tone which could be factual or non- factual. |
| ***Imperative*** | These clauses are in the form of a request, command, rule, or advice. |
| ***Relative*** | Simply it is a subordinate clause. |
| ***Interrogative*** | Clauses in a form of questions. |
| ***Exclamative*** | These clauses are used to express shock, surprise, or even compliments. these clauses often end with an exclamation mark. |

*Table 3 Clause Categories*

## 3.4. Grammar

In linguistics, grammar is the set of structural rules governing the composition of clauses, phrases, and words in any given natural language. grammar is not just a fixed set of rules but also its evolution based on the usage of language over time among humans.

Grammar can be subdivided into two main classes dependency grammars and constituency grammars based on their representations for linguistic syntax and structure.

### 3.4.1. Dependency grammars

Dependency grammar (DG) is a class of modern syntactic theories that are all based on the dependency relation (as opposed to the constituency relation). Dependency is the notion that linguistic units, e.g. words, are connected to each other by directed links. The (finite) verb is taken to be the structural center of clause structure. All other syntactic units (words) are either directly or indirectly connected to the verb in terms of the directed links, which are called dependencies. DGs are distinct from phrase structure grammars (constituency grammars). Consequently, dependency grammars assume that further constituents of phrases and clauses are derived from this dependency structure between words.

Dependency grammar characterized by root word that has no dependency to other words, all other words has some relationship or dependency on other words.

The verb is taken as the root of the sentence in most cases. Although there are no concepts of phrases or clauses, looking at the syntax and relations between words and their dependents, one can determine the necessary constituents in the sentence.

Dependency grammars always have a one-to-one relationship correspondence for each word in the sentence. There are two aspects to this grammar representation. One is the syntax or structure of the sentence, and the other is the semantics obtained from the relationships denoted between the words. The syntax or structure of the words and their interconnections can be shown using a sentence syntax or parse tree similar to that depicted in an earlier section.

Dependency grammars has two aspects representation:

- Syntax or structure of the sentence.
- Semantics of the sentence: is the meaning obtained from the relationships between the words.

Parse tree is used to represent the syntax or structure of the words and their interconnections. As an example our sentence ['Anna kills wild wolf and takes it away'], the dependency graph generated by Stanford parser and graphviz: a graph generation tool is shown in figure [4].

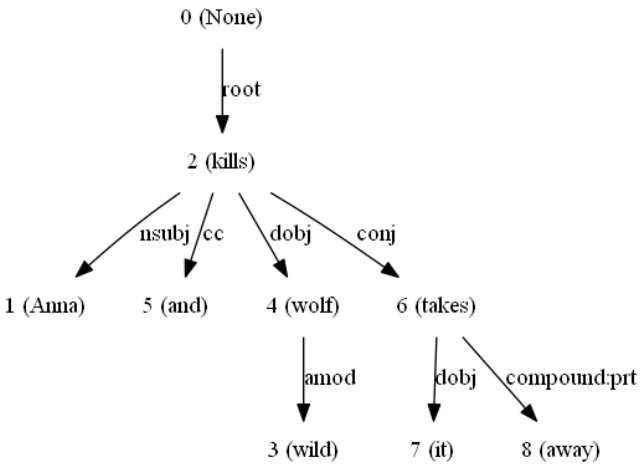



**Figure 4 Dependency Syntax Grammar generated by Stanford Parser**

Dependency tree is a directed acyclic graph (DAG). That is, it is formed by a collection of vertices (Words) and directed edges, each edge connecting one word to another, such that there is no way to start at

some word W and follow a sequence of edges that eventually loops back to W again. Words order is not depicted by dependency trees but it depict more the relationship between the words in the sentence.

We can note that different variations of graph can be generated to the same sentence based on the parser we use e.g. Stanford dependency parser 2014 generate different graph than parser of 2018 and this variation depends on parser training methodology and dataset it uses for training and the annotator it uses for POS tagging.

The Stanford typed dependencies designed to provide a simple description of the grammatical relationships in a sentence. In particular, rather than the phrase structure representations that have long dominated in the computational linguistic community, it represents all sentence relationships uniformly as typed dependency triples relations between pairs of words. There are 55 dependency Relation defined by Stanford as in table [4].

| Dependency Relation | Description |
|---|---|
| **dep – dependent** | A dependency is labeled as dep when the system is unable to determine a more precise dependency relation between two words. |
| **aux – auxiliary** | An auxiliary of a clause is a non-main verb of the clause, |
| **auxpass - passive auxiliary** | A passive auxiliary of a clause is a non-main verb of the clause which contains the passive information. |
| **cop – copula** | A copula is the relation between the complement of a copular verb and the copular verb. |
| **agent – agent** | An agent is the complement of a passive verb which is introduced by the preposition “by" and does the action. |
| **comp – complement** | A complementizer of a clausal complement (ccomp) is the word introducing it. It will be the subordinating conjunction “that" or “whether". |
| **acomp – adjectival Complement** | An adjectival complement of a verb is an adjectival phrase which functions as the complement (like an object of the verb). |
| **attr – attributive** | An attributive is a WHNP complement of a copular verb such as “to be", “to seem", “to appear". |
| **ccomp - clausal complement with internal subject** | A clausal complement of a verb or adjective is a dependent clause with an internal subject which functions like an object of the verb, or adjective. |
| **xcomp - clausal complement with external subject** | An open clausal complement (xcomp) of a VP or an ADJP is a clausal complement without its own subject, whose reference is determined by an external subject. |
| **compl – complementizer** | A complementizer of a clausal complement (ccomp) is the word introducing it. |
| **dobj - direct object** | The direct object of a VP is the noun phrase which is the (accusative) object of the verb. |
| **iobj - indirect object** | The indirect object of a VP is the noun phrase which is the (dative) object of the verb. |
| **pobj - object of preposition** | The object of a preposition is the head of a noun phrase following the preposition, or the adverbs “here" and “there". |
| **rel - relative (word introducing a rcmod)** | A relative of a relative clause is the head word of the WH-phrase introducing it. |

| | |
|---|---|
| **nsubj - nominal subject** | A nominal subject is a noun phrase which is the syntactic subject of a clause. |
| **nsubjpass - passive nominal subject** | A passive nominal subject is a noun phrase which is the syntactic subject of a passive clause. |
| **csubj - clausal subject** | A clausal subject is a clausal syntactic subject of a clause, i.e., the subject is itself a clause. |
| **csubjpass - passive clausal subject** | A clausal passive subject is a clausal syntactic subject of a passive clause. |
| **cc – coordination** | A coordination is the relation between an element of a conjunct and the coordinating conjunction word of the conjunct. |
| **conj – conjunct** | A conjunct is the relation between two elements connected by a coordinating conjunction, such as “and", “or". |
| **expl - expletive (expletive there")** | This relation captures an existential “there". The main verb of the clause is the governor. |
| **abbrev - abbreviation modifier** | An abbreviation modifier of an NP is a parenthesized NP that serves to abbreviate the NP (or to dene an abbreviation). |
| **amod - adjectival modifier** | An adjectival modifier of an NP is any adjectival phrase that serves to modify the meaning of the NP. |
| **appos - appositional modifier** | An appositional modifier of an NP is an NP immediately to the right of the first NP that serves to dene or modify that NP. |
| **advcl - adverbial clause modifier** | An adverbial clause modifier of a VP or S is a clause modifying the verb. |
| **purpcl - purpose clause modifier** | A purpose clause modifier of a VP is a clause headed by “(in order) to" specifying a purpose. |
| **det – determiner** | A determiner is the relation between the head of an NP and its determiner. |
| **predet – predeterminer** | A predeterminer is the relation between the head of an NP and a word that precedes and modifies the meaning of the NP determiner. |
| **preconj – preconjunct** | A preconjunct is the relation between the head of an NP and a word that appears at the beginning bracketing a conjunction |
| **infmod - innitival modier** | An innitival modifier of an NP is an innitive that serves to modify the meaning of the NP. |
| **partmod - participial modier** | A participial modifier of an NP or VP or sentence is a participial verb form that serves to modify the meaning of a noun phrase or sentence. |
| **advmod - adverbial modier** | An adverbial modifier of a word is a (non-clausal) adverb or adverbial phrase (ADVP) that serves to modify the meaning of the word. |
| **neg - negation modier** | The negation modifier is the relation between a negation word and the word it modifies. |
| **rcmod - relative clause modier** | A relative clause modifier of an NP is a relative clause modifying the NP. The relation points from the head noun of the NP to the head of the relative clause, normally a verb. |
| **quantmod - quantier modier** | A quantifier modifier is an element modifying the head of a QP constituent. |
| **tmod - temporal modier** | A temporal modifier (of a VP, NP, or an ADJP is a bare noun phrase constituent that serves to modify the meaning of the constituent by specifying a time |
| **nn - noun compound modifier** | A noun compound modifier of an NP is any noun that serves to modify the head noun. |

| | |
|---|---|
| **num - numeric modifier** | A numeric modifier of a noun is any number phrase that serves to modify the meaning of the noun. |
| **number - element of compound number** | An element of compound number is a part of a number phrase or currency amount. |
| **prep - prepositional modier** | A prepositional modifier of a verb, adjective, or noun is any prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another preposition. |
| **poss - possession modifier** | The possession modifier relation holds between the head of an NP |
| **possessive - possessive modifier ('s)** | The possessive modifier relation appears between the head of an NP and the genitive 's. |
| **prt - phrasal verb particle** | The phrasal verb particle relation identifies a phrasal verb, and holds between the verb and its particle. |
| **parataxis – parataxis**<br>**punct – punctuation** | This is used for any piece of punctuation in a clause, if punctuation is being retained in the typed dependencies. |
| **xsubj - controlling subject** | A controlling subject is the relation between the head of open clausal complement (xcomp) and the external subject of that clause. |
| **Root** | The root grammatical relation points to the root of the sentence. |

**Table 4 Stanford Typed Dependency Relations [9]**

### 3.4.2. Constituency Grammars.

Constituency grammars is a set of grammars that derives constituents of the sentence by represent its hierarchically ordered structure. a constituent is a word or a group of words that function(s) as a single unit within a hierarchical structure. The analysis of constituent structure is associated mainly with phrase structure grammars that utilize a set of rules and syntax that govern the hierarchy and ordering of sentence constituents. phrase structure grammars have two tasks: first identify words of phrase or constituents then apply the rules to identify its order.

The analysis of phrase structure depicted by the generic representation of a phrase structure rule pattern as:

| **Structure pattern** | **Intended for** |
|---|---|
| **S—> AB** | *S:* structure<br>A, B: are constituent's *A* and *B*<br>*Order is: A followed by B* |

Rules includes, breaking down sentence or clause into its constituents. E.g. ***S → NP VP*** the first level is *noun phrase* (NP) and *verb phrase* (VP). First level components can be divided into smaller size one by apply another rules for example the rule for representing a noun phrase is ***NP → [DET][ADJ]N [PP],*** where [] is an optional parameter and N is the head. various constituents a noun phrase typically contains as in Table [5].

| | | | |
|---|---|---|---|
| **NP→N** | **NP→DET N** | **NP→DET ADJ N** | **NP→NP PP** |

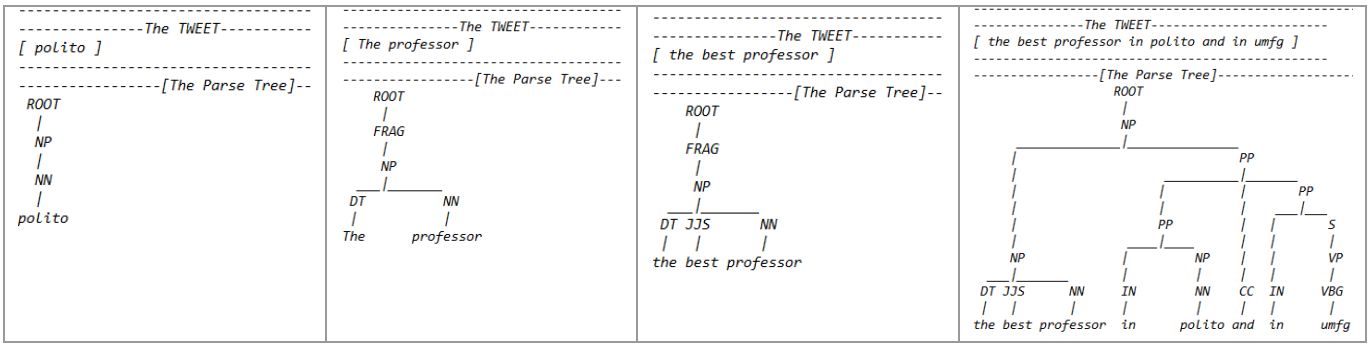



**Figure 5 Noun Phrase rules for Constituency trees Representation**

Verb construction rule is: ***VP → V | MD [ V] [ NP] [ PP] [ ADJP] [ ADVP]***, where the head word is usually a verb (V) or a modal (MD). figure [6] show various constituents in verb phrases,

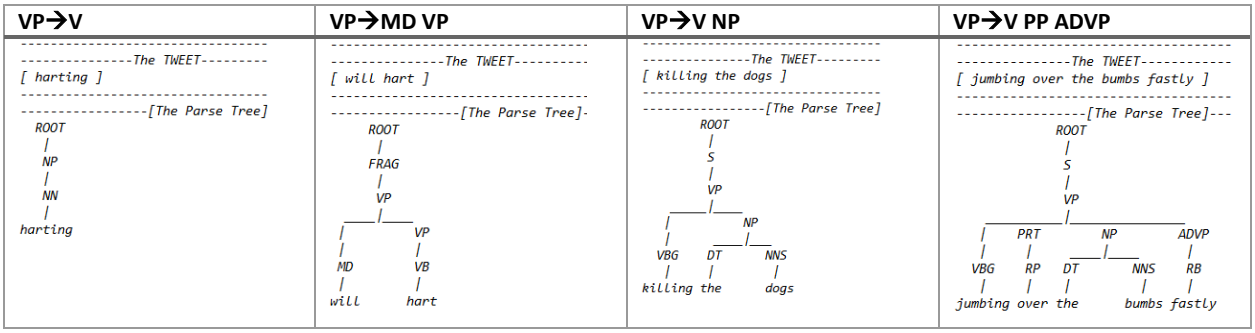



**Figure 6 Verb Phrase rules for Constituency trees Representation**

Verb Phrase may also contain another verb phrase, Noun phrase or adverbial phrase. The production rules for representing prepositional phrases is: ***PP → PREP [ NP]***, where PREP is a preposition acts as the head some examples in figure [7].

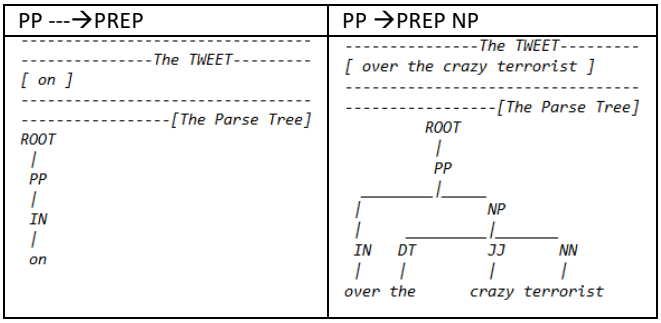



**Figure 7 Prepositional Phrase rules for Constituency trees Representation**

Recursion is an inherent important property of language that allows constituents to be embedded in other constituents and that the production rules are inherently applicable as we see in example “The hatting people in the city on the hill by their sentiment” in figure [8].

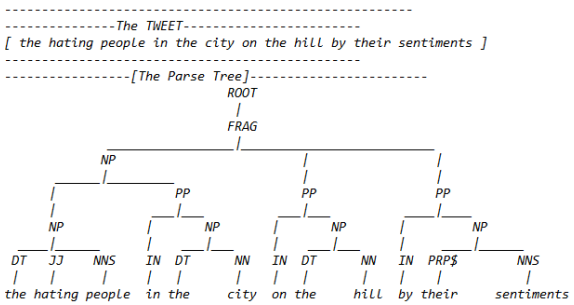



**Figure 8 . Constituency tree utilize recursive nested properties of NP and PP**

Two sentences or clause can be joined by conjunctions. The production rule can be denoted as ***S*** → ***S conj S***. *example in figure [9] show two noun phrases joined by conjunction.*

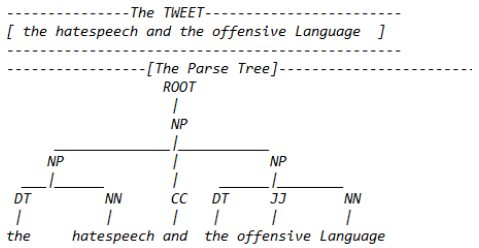



**Figure 9 Constituency tree of two NP joined by a conjunction**

Moreover, the constituency grammar–based production rules break down the top-level constituents into further constituents as in figure [10].

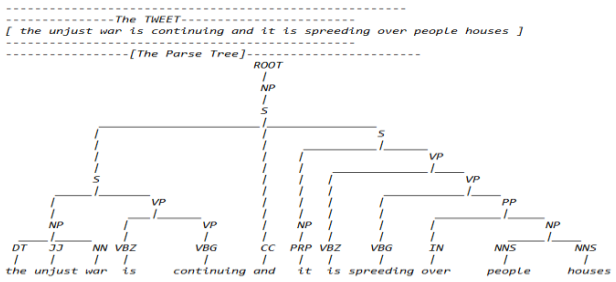



**Figure 10 Constituency tree top sentence break down into two sentences**

# CHAPTER 4.

# (LINGUSTIC FRAMEWORK AND PARSERS).

In this section, we will describe different kind of parsers build based on the phenomenon in the previous section and it works as a base of our work in (Natural Language Structure Extraction).

## 4.1. Parser Types

There is different type of parsers in NLP figure [11] sow some of them.

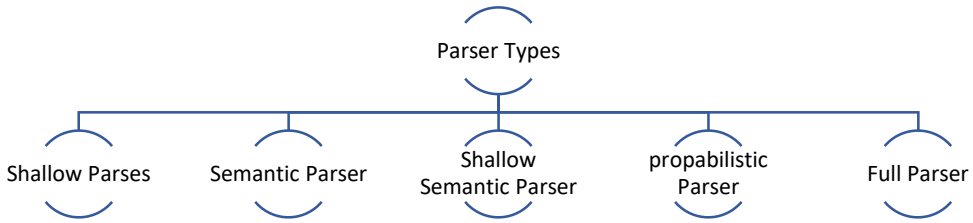



*Figure 11 NLP Parsers Type*

- Shallow Parser
- Semantic Parser
- Shallow Semantic Parser
- Probabilistic Parser
- Full Parser

### 4.1.1. Shallow Parser:

Shallow parsing also called (chunking, "light parsing") is an analysis that identifies the constituents of the sentence as Noun or Verb groups or phrases etc, in this parsing neither internal structure nor their role in the main sentence are specified It is a technique widely used in natural language processing. Shallow Structure Hypothesis explain the reasons why second language learners often fail to parse complex sentences correctly.[22]

This parser outputs only the syntactical information by chunking. It is similar to the concept of lexical analysis for computer languages[36]. As in example of figure [12].

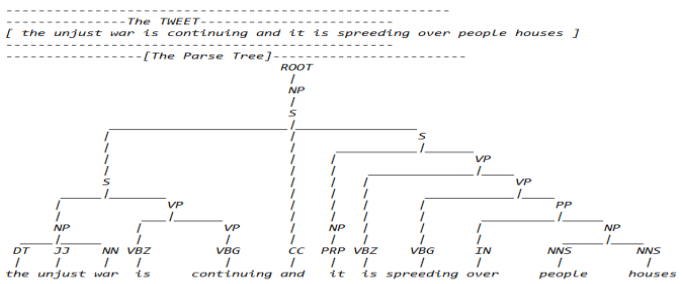



*Figure 12 shallow parsing output of 'The unjust war is continuing and it is spreading over people houses'*

### 4.1.2. Semantic Parser:

In linguistics, semantic parsing is the process of relating syntactic structures, from the levels of phrases, clauses and sentences to the level of the writing as a whole to match the text with a formal meaning representation. Stanford have developed a Semantic Parser called a Stanford's SEMPRE Parser based on supervised and un-Supervised Machine Learning.

### 4.1.3. Shallow Semantic Parser:

Looking for the predicate and find its complement by answering some questions related to the Actor, target, timing and positioning as Who, Where, Which, Where, Why, etc.

### 4.1.4. Probabilistic Parser:

This parser is build based on training and predicting in the parsing process to improve its estimated output. An example of this parser is Stanford NLP group's probabilistic parser called Stanford Parser.

### 4.1.5. Full Parser:

Is the combination of all the previous parsers on the parsing stack, it is based on shallow semantic parsing and probabilistic parsing, the output of this parsing is the dependency Parse relation between different components of a sentence, the base of this parser by applying the rules introduced on the section of Dependency Grammar, this parser is the core of our Design of Heuristic Algorithms for Natural Language Linguistic Structure Feature Extraction for Hate Speech and offensive language classification.

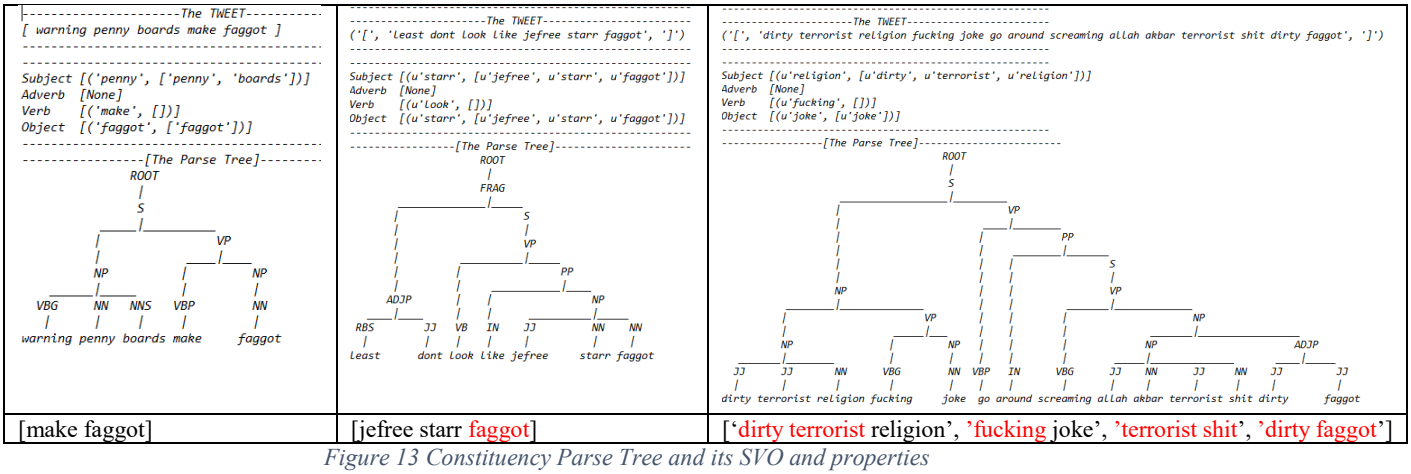
# CHAPTER 5.

# (STRUCTURED BASD SYSTEM ANALYSIS).

Basically, Supervised learning is the machine learning task of inferring a function from labeled training data. [37]. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new testing data. The accuracy of this system depends on two parts, first the labeled dataset accuracy annotated by human experts, the second is the feature generated from the data field of the dataset usually the main Problem of the supervised Learning in Natural Language Processing is that features generated from dataset are in ambiguities way contains for relevant and non-relevant data to the domain of interest and this feature is the training content of the system, and the accuracy of such classifier system depend on the significant information and terms existed with in the provided learning dataset, although features can be a set that directly represent the dataset as the raw tweet text or indirectly represent the dataset as sentimental feature of the tweets or it may be a meta data that try to extract the user behavior, those feature converted to a vector set and represented by numerical values to be encoded for recognition.

The key idea her is that if we identify precisely the direct relevant terms and the relationship between them eliminating non relevant data from our focus have an impact on the system accuracy and that as much as we understand the meaning of the terms and its relation forming these relations in a way that understood by the underlain machine learning system the detection accuracy will converges to 100%.

So, we made an analysis of 1700 tweet by showing the hierarchical Grammatical structure and generating the Constituency Parse tree of those tweets and identifying the significant parts of the tweets that contains the hate and or offensive terms and the relation between them as we can see in figure [13] the output of our analysis of three kind of tweets messages:




| | | |
|---|---|---|
| [make faggot] | [jefree starr faggot] | [‘dirty terrorist religion’, ’fucking joke’, ’terrorist shit’, ’dirty faggot’] |

*Figure 13 Constituency Parse Tree and its SVO and properties*

On the first example the tweet “ warning penny boards make faggot”, the analysis result show that the hate or offensive term “make faggot” concentrated with in the verb Predicate VP, on the Second example “least don’t look like jefree starr faggot” the hate terms concentrated with in the Noun Predicate “jefree starr

faggot" and can be shown as ["adj Noun (JJ NN)" or "Noun Noun (NN NN)"] in witch the faggot Noun modified by the terms "jefree starr" as its adjective, in other way we can say that "fagot" is used to describe the Noun "starr", so the structure [<N N> ,<jj N>,<NP>] is added to the template structure list we will use in the Design stage of the Heuristic algorithms ,on the third example the hate terms ['dirty terrorist religion'] "dirty terrorist <jj jj> " as an adjective describe other adjective and "terrorist religion <jj NN>" as an adjective 'terrorist' describe or modify the name 'religion' these consonants grouped as Noun Predicate (NP), in the same way 'fucking joke <VP>' is a verbal phrase that is used to describe the previous noun Predicate and is a part of the higher level Noun predicate as we can see in figure [13] , 'terrorist shit<jj NN> is a NP' and 'dirty faggot <jj jj>' an adjective predicate.

we can see in the last tweet how the twitter-er (user) express his sentimental hate expression on many phrasal components that show the extent and magnitude of the hate level and this magnitude must be transformed and added to features used in the machines learning system, so in this earlier step of the analysis all phrases are added to the list of noted hate structure.

The rule of either to add the template to the final list or note is that if this template has been noted frequently on the analyzed 1700 tweets.

On this analysis, we used the output of the shallow parsing to extract the [Subject Verb Object Adverb] by traversing up from the root on the left and right branches of the root to extract the subject and Object. but the result is not so accurate by since subject, verb and object can be separated by words, subject may be in the same sentence or in the previous sentences in the same way the object can be in this same sentence or in the following one, in this regards the dependency grammar can help with inferring this relationship among the words, we will see how we have applied a set of rules to extract each structure template later on the design chapter.

The output of this analysis is a list of frequently used template structure that contains for the significant parts of the hate and offensive language, we will develop a set of heuristic algorithms using some rules to extract these templates, The following "Template Structure for Hate Speech detection in English Language "result from our analysis:

| **Template Structure** |
|---|
| **{**<br>**[SVO (subject Verb Object)],[<Subj><Adv><Verb><Adj><Obj>],**<br>**[<Subj><Adv><Verb><Obj>] ,[<Subj><Verb><Adj><Obj>], [<Adj Predicate>], [<N Predicate>], [<V Predicate>], [<mix Adj V N>], [<Subj><Adj><Obj>], [<Adj><Obj>], [<Adj><Noun>]**<br>**}** |

Where the <Adv> <Verb> <Adj> Is a Hate Terms represented in different structure, And That <Subj> <Obj> is an ENTITY representing the Hate source and target, we noted that the second template [<Subj><Adv><Verb><Adj><Obj>] used by some researchers to express the hate speech intensity and intent of the users [I< intensity > <userintent > < hatetarget >].

Our goal of this analysis to generalize the detection of hate and offensive expression expressed with different syntactic structure and context.

The main goal of structure based system analysis is the transformation of tweets into the linguistic structure, as a hierarchical grammatical form and in order to create a system which understands a natural language, by development of OIE (Open Information Extraction) system for Linguistic Extraction. therefor parsing process is needed to transform Natural Language to the form that show its morphological description

together with its dependency relation, to form the linguistic framework of the system, at this point we need to pass the parsed data for farther analysis to generate a more relation between sentences components by setting some rules and assumptions on top of dependency parsed trees for features generation.

# CHAPTER 6.

# (HEURISTICS ALGORITHMS DESIGN AND DEVELOPMENT).

With Reference to Analysis have been used before, we have concluded a set of template structure as a result of our analysis stage, we will develop a system able to extract these relations list by parsing the sentences using Stanford dependency parser.

Stanford has designed a typed dependencies representation to provide a description of the grammatical relationships in a sentence based on the linguistics grammatical theories we described on analysis chapter. In particular, rather than the phrase structure representations, it represents all sentence relationships uniformly as typed dependency relations. That is, as triples of a relation between pairs of words. The result relation represented as a triples as the following:

**PARSE TEXT BY STANFORD DEPENDENCY PARSER AND RETURN THE TRIPLES OF**

**Dependency List= [**
**((word,POS),Dependency_Relation,(word,POS))**
**((word,POS),Dependency_Relation,(word,POS))**
**..**
**((word,POS),Dependency_Relation,(word,POS))**
**]**

The current representation of Stanford typed dependency contains 55 grammatical relations [73]. The dependencies are all binary relations: a grammatical relation holds between a governor and a dependent. The definitions make use of the Penn Treebank part-of-speech tags and phrasal labels.

Upon Stanford dependency relation parsed tree we have introduced a set of assumptions and rules on top of dependency parsed triples to identify the parties of the templates structure for extrications. Those rules and assumptions is the base of development of Heuristic algorithms capable for extract the Linguistic meaning of sentences in terms of pairs and triples that reflect the real life activities and sentiments so that sentence can be represented by a set of structural relations among the significant entities of the sentence. Those composed relations contribute on the final formation of the main concept and meaning:

We have formulated assumptions and manipulation rules on top of Dependency tree for each type of relation extraction, in the following sections we will describe the rules and assumptions for each templates structure extraction and the manipulation algorithms. In chapter 4 we conclude a set of templates to be extracted by our developed algorithms.

In linguistics, ***Word Order Typology*** is the study of the language constituent's syntactic orders, and how different languages can employ different orders. The primary word orders that are of our interest are the constituent order of a clause as Subject, Verb and Object. In English, the frequency distribution of word order research's surveyed by Russell S. Tomlin in 1980s [68] [69] show that SVO word order has a frequency of 42% on proportions of other languages as we see in table [8] below.

So, according to this order of English Language we follow the SVO order to implement a heuristics algorithm for this template extraction.

| Word Order | Example | Language frequency | Languages |
|---|---|---|---|
| **SOV** | "She him loves." | 45% | Hindi, Latin, Japanese, Korean, Marathi |
| **SVO** | "She loves him." | 42% | English, Hausa, Mandarin, Russian |
| **VSO** | "Loves she him." | 9% | Biblical Hebrew, Irish, Filipino, Tuareg |
| **VOS** | "Loves him she." | 3% | Malagasy, Baure |
| **OVS** | "Him loves she." | 1% | Apalaí, Hixkaryana |
| **OSV** | "Him she loves." | 0% | Warao |

*Table 5 Frequency distribution of word order in languages*

In the following section we will introduce a set of rules and assumptions on locating parts of structure of the dependency tree:

## 6.1. Subject Extraction.

The simple definition of the subject is the person or thing about whom the statement is made, but for complex statements subject can be defined according to different traditions and different technical perspective as in table [9]:

| Subject | According To | Definition |
|---|---|---|
| **Def.1** | Simple statements | Is the person or thing about whom the statement is made. |
| **Def.2** | According to a tradition that refers to Aristotle that is associated with phrase structure grammars. | Is one of the two main constituents of a clause, the other constituent being the predicate that describe the subject (governor or clause modifier). [70][71]. |
| **Def.3** | According to predicate logic and dependency grammars. | Is the most prominent overt argument of the predicate. [72] |
| **Def.4** | from a functional perspective | is a phrase that conflates nominative case with the topic. |

*Table 6 Subject Definitions*

### 6.1.1. Subject Manipulation Rules and Assumptions:

Assumption 1. Subject is the noun modified by the relative clause modifiers, the relative clause that modified the subject (the head of the relation) can be a verb targeted to the subject direct or indirect way. This assumption according to definition 2 of subject table [9]

Assumption 2. Subject is a noun or noun phrases that represented as the source of the action on the clause, the clause can be active or passive.

Assumption 3. In passive clause the subject is not necessary to be followed by the former verb and the clause is formed by making a transformation using the associated auxiliary verb and past participle and in this case the subject can be obtained by a reverse transformation to its origin.

Rule 1. The root of the DP is either a non-copular verb or the subject complement of a copular verb. Starting at the head of Dependency parse tree the Subject relation can be (Nsubj, Nsubjpass, Xsubj, Csubj or Csubjpass) and the governor can be a direct verb, a passive verb or a clause denoted as the second parameter.

Rule 2. If the subject relation has a WH- POS type, then the subject can be either the WH word or the relative clause denoted by WH word.

According to subject definitions in table [9], subject is a constituent that can be realized in numerous forms table [10] show different forms of the subject defined by Stanford typed Dependency Relation [73]:

| Rules Applied | Dependency Relation | Definition and Example |
|---|---|---|
| **Assumption 2 Rule 1** | **Nominal Subject** (Nsubj) | A nominal subject is a noun phrase which is the syntactic subject of a clause. The governor of this relation might not always be a verb: when the verb is a popular verb, the root of the clause is the complement of the copular verb, which can be an adjective or noun.<br>E.g. "Clinton defeated Dole" nsubj(defeated, Clinton)<br>"The baby is cute" nsubj(cute, baby) |
| **Assumption 3 Rule 1** | **Passive Nominal Subject** (Nsubjpass) | A passive nominal subject is a noun phrase which is the syntactic subject of a passive clause.<br>E.g "Dole was defeated by Clinton" nsubjpass (defeated, Dole) |
| **Assumption 1 Rule 1** | **Controlling Subject** (Xsubj) | A controlling subject is the relation between the head of a open clausal complement (xcomp) and the external subject of that clause. This is an additional dependency, not a basic dependency.<br>E.g. "Tom likes to eat fish" xsubj(eat, Tom) |
| **Assumption 2,3 Rule 1** | **Clausal Subject** (Csubj) | A clausal subject is a clausal syntactic subject of a clause, i.e., the subject is itself a clause. The governor of this relation might not always be a verb: when the verb is a copular verb, the root of the clause is the complement of the copular verb. In the two following examples, "what she said" is the subject.<br>E.g. "What she said makes sense" csubj(makes, said)<br>"What she said is not true" csubj(true, said) |
| **Assumption 2,3 Rule 1** | **Clausal Passive Subject** (Csubjpass) | A clausal passive subject is a clausal syntactic subject of a passive clause. In the example below, "that she lied" is the subject.<br>E.g. "That she lied was suspected by everyone" csubjpass(suspected, lied) |
| **Assumption 1 Rule 2** | **Relative clause Modifier** (Acl:relcl) | Is a relative clause modifying the Noun, the relation points from the noun that is modified to the head of relative clause.<br>E.g. I saw the man you love Acl:relcl( man, love)<br>I found the book which you bought Acl:relcl(book, bought) |

*Table 7 Subject Forms (Stanford Typed Dependency) [73]*

### 6.1.2. Subject Extraction Algorithm.

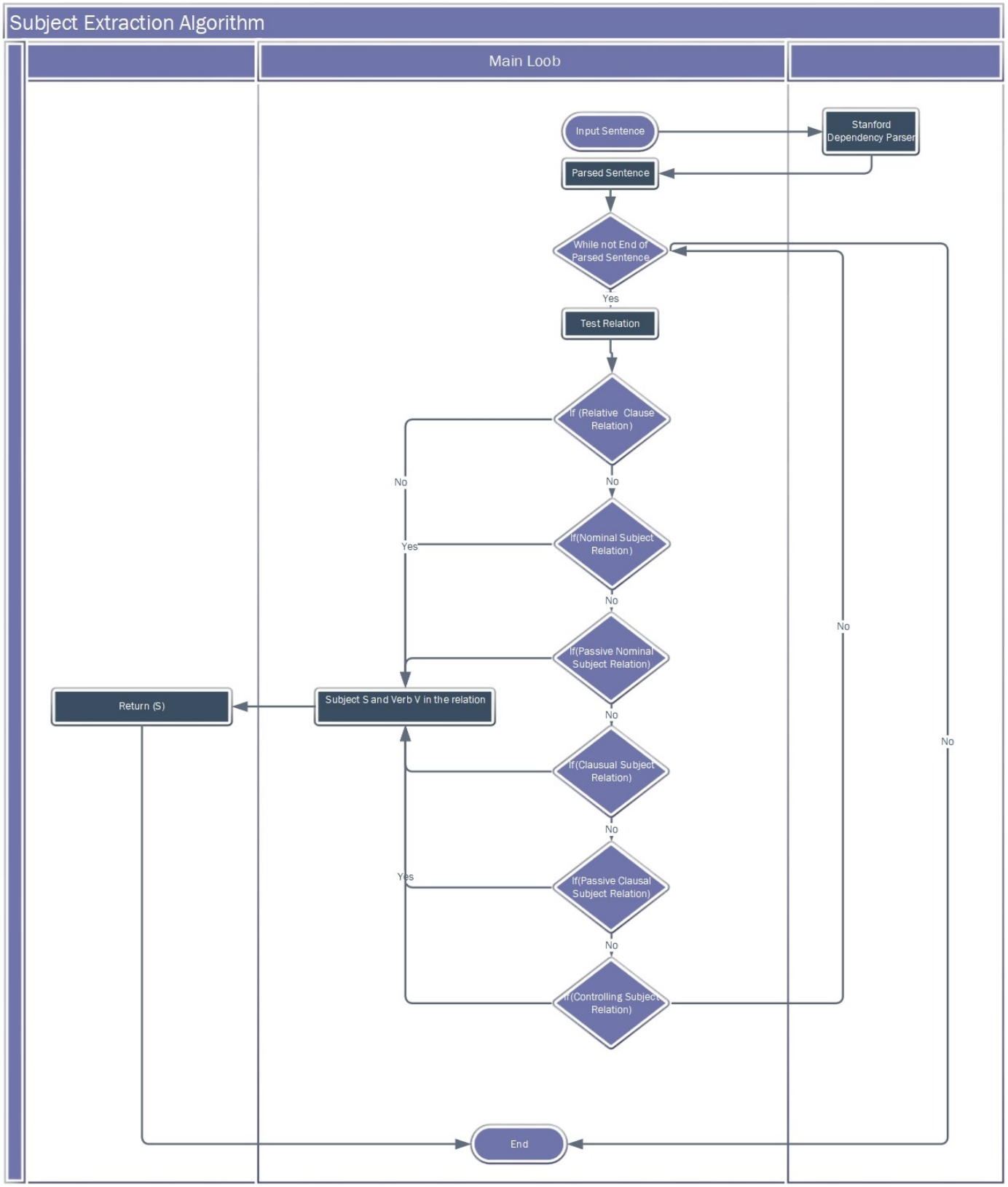



*Figure 14 Subject Detection Algorithm*

## 6.2. Verb Extraction.

A verb, is a word (POS) that conveys an action (write, listen, run), an occurrence (happen, become), or a state of being (be, exist, stand). the basic form is the infinitive. It is modified in form to encode tense, aspect, mood, and voice. A verb may also agree with the person, gender, and/or number of some of its arguments, such as its subject, or object. Verbs have tenses.

Verb is the active part of the dependency relation and has a largest number of relations with subject, auxiliary verbs, other verbs and different type of objects.

### 6.2.1. Verb Forms.

Verbs vary by type, type determined by the words relation to the verb. There are three basic types: intransitives, transitive's, ditransitive and double transitive verbs. Some verbs have special grammatical uses and hence complements, such as copular verbs, verb "do", auxiliaries verbs and it can have various special forms such as infinitives, participles or gerunds. [74]

| Verb Form | Description |
|---|---|
| **Transitive verbs** | A transitive verb is verb followed by a noun or noun phrase as direct object. When two noun phrases follow a transitive verb, the first is an indirect object, that which is receiving something, and the second is a direct object, that being acted upon. Indirect objects can be noun phrases or prepositional phrases |
| **Ditransitive verbs** | Ditransitive verbs precede either two noun phrases or a noun phrase and then a prepositional phrase often led by to or for. For example: "The players gave their teammates high fives.". [75] |
| **Intransitive verbs** | An intransitive verb is one that does not have a direct object. Intransitive verbs may be followed by an adverb. For example: "The woman spoke softly." |
| **Double transitive verbs** | Double transitive verbs are followed by a noun phrase that serves as a direct object and then a second noun phrase, adjective, or infinitive phrase. The second element (noun phrase, adjective, or infinitive) is called a complement. For example: "The young couple considers the neighbors wealthy people. |
| **Copular verbs** | Copular verbs must be followed by a noun or adjective, whether in a single word or phrase. Copulae are thought to 'link' the adjective or noun to the subject.<br>Adjectives that come after copular verbs are predicate adjectives, and nouns that come after linking verbs are predicate nouns. [76] |

*Table 8 Verb Forms*

processing dependency parse tree for verb examination is done so that verbs usually presented as the root of the tree as we described on dependency grammar, so that verbs annotated by POS tags as in table [12] bellow is detected as verb it refers to one of basic verbs form as in table [11].

| POS | Meaning |
|---|---|
| **VB** | Verb, base form |
| **VBD** | Verb, past tense |
| **VBG** | Verb, gerund or present participle |
| **VBN** | Verb, past participle |
| **VBP** | Verb, non-3rd person singular present |
| **VBZ** | Verb, 3rd person singular present |

*Table 9 POS Verb Tags*

After the subject have been detected the second part of the typed dependency relation can be either a verb or a compliment of the clause. If verb have been detected the next step is to traverse the tree looking for the object.

### 6.2.2. Verb Extraction Algorithm.

Verb Extraction is not an independent task it is embedded by subject and object extraction Algorithm and it refers to the second argument of the typed dependency relations on most of tested relations. Output of Subject Extraction Algorithm and object Extraction Algorithm is a pairs of "S+V" and "V+O" respectively.

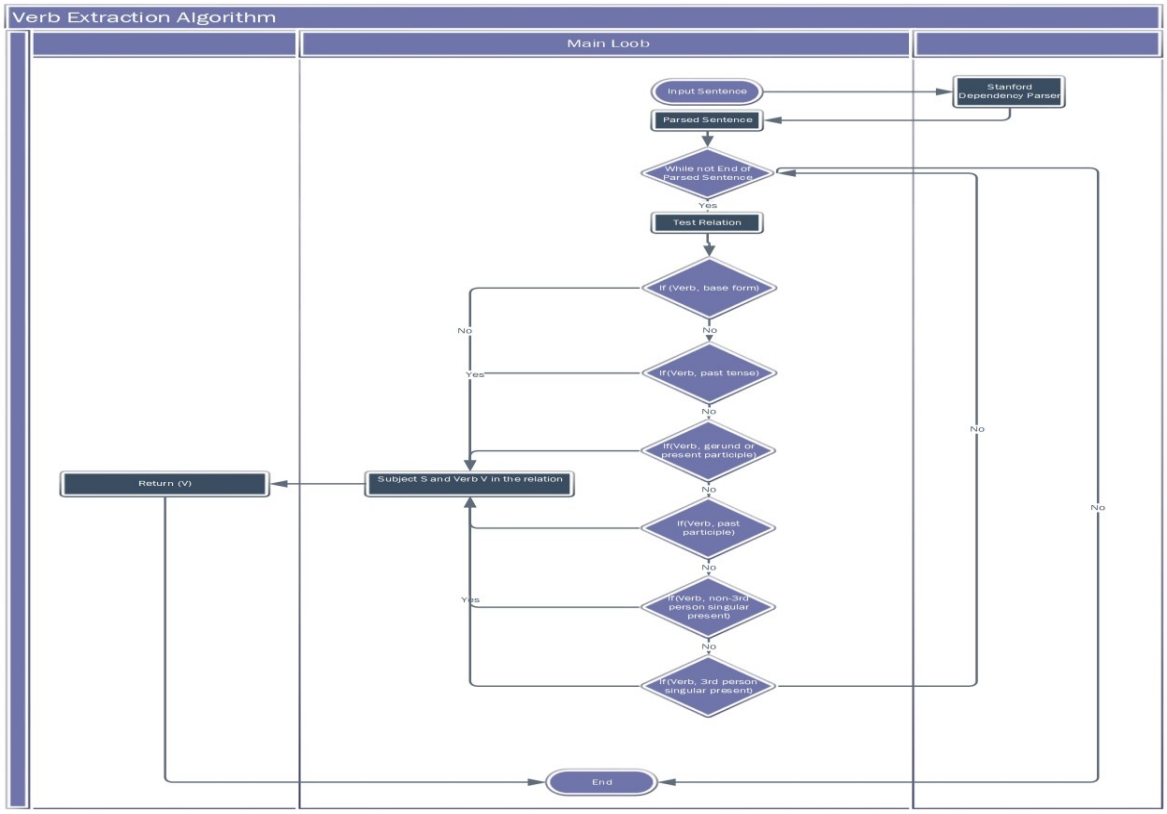



*Figure 15 Verb Extraction Algorithm*

## *6.3. Object Extraction.*

Traditional grammar defines the object in a sentence as the entity that is acted upon by the subject. [77] Traditional theories of sentence structure divide the simple sentence into a subject and a predicate, [78] whereby the object is taken to be part of the predicate. [79] dependency grammar theories, define the object to be as the verb arguments. [80]

### 6.3.1. Object Manipulation Rules and Assumptions.

Rule 1. Object can be presented or not based on the type of the main verb in the clause.
   a. Transitive verbs require an object.
   b. intransitive verbs don't require an object. [81]
   c. The object can be the complement of the clause but not always.

Rule 2. Object can be a noun or noun phrases followed the verb or a nominal noun or noun phrases that works as an attributes of it.

Rule 3. In a sentence that contains an auxiliary verb, the object is the attributes follow the verbs of the

auxiliary that can be a noun, noun phrase, adjective or an adjective phrase. The auxiliary obtained by testing the relation ('aux' or 'auxpass')

Rule 4. Object can be clausal complement of a VP or an ADJP usually came after "to" in this case the object is the object of the second verb of the clausal complement e.g. the object of the verb after "to" e.g. I think that you like (to eat) at restaurant (restaurant is the object of the second verb "to eat"). This complement identified by testing the ('xcomp') relation.

Rule 5. In Relative clause in witch noun phrases have modified by relative clause. we replace the relative pronoun (e.g., who or which) of a relative clause by its antecedent obtained By ('rcmod') dependency to the governor of the relative pronoun.

Rule 6. In general, direct object is the entity acted upon and it is identified by testing the dependency relation ('dobj').

Rule 7. Indirect object identified as an entity indirectly affected by the action, it is identified by testing the dependency relation ('iobj').

Rule 8. Object can be a target of the proposition. Table [13] show an example of those three different types. It can be identified by testing the ('pobj') relation.

Rule 9. In case of two clauses linked by conjunction if the conjunction links between a Noun and the subject of the clause then looking for object done by a recurrent search by replacing the Noun by the Subject in the new searched clause e.g. "He drink water and juice" the clause splitting will be "He drink water", "He dink Juice" so objects will be [Water and Juice].

| Type | Description | Example |
|---|---|---|
| **Direct object** | Entity acted upon | Sara fed **the dogs**. |
| **Indirect object** | Entity indirectly affected by the action | She sent **him** a present. |
| **Prepositional object** | Object introduced by a preposition | She is waiting **for Tommy**. |

*Table 10 General object types*

On the following table [14] a set of dependency relations used to locate the object with in the clause by apply the rules and assumptions defined before.

| Rules | Dependency Relation | Description |
|---|---|---|
| **Rule 3** | aux: auxiliary | An auxiliary of a clause is a non-main verb of the clause, e.g. modal auxiliary, "be" and "have" in a composed tense.<br>e.g. Reagan has died" aux(died, has)<br>He should leave" aux(leave, should) |
| **Rule 3** | auxpass: passive auxiliary | A passive auxiliary of a clause is a non-main verb of the clause which contains the passive information.<br>e.g. Kennedy has been killed" auxpass(killed, been) aux(killed,has)<br>Kennedy was got killed" auxpass(killed, was/got) |
| **Rule 9 And any of Applicable Rules [1-9]** | conj : conjunct | A conjunct is the relation between two elements connected by a coordinating conjunction, such as "and", "or", etc. We treat conjunctions asymmetrically: The head of the relation is the first conjunct and other conjunctions depend on it via the conj relation.<br>e.g. "Bill is big and honest" conj (big, honest)<br>"They either ski or snowboard" conj (ski, snowboard) |
| **Rule 3 with Different** | cop: copula | A copula is the relation between the complement of a copular verb and the copular verb. (We normally take a copula as a dependent of its complement; see the discussion in section 4.) |

| | | |
|---|---|---|
| **Order** | | e.g. Bill is big" cop(big, is)<br>Bill is an honest man" cop(man, is) |
| **Rule 6** | dobj : direct object | The direct object of a VP is the noun phrase which is the (accusative) object of the verb.<br>e.g. She gave me a raise" dobj (gave, raise)<br>They win the lottery" dobj (win, lottery) |
| **Rule 7** | iobj : indirect object | The indirect object of a VP is the noun phrase which is the (dative) object of the verb.<br>e.g. She gave me a raise" iobj (gave, me) |
| **Rule 8** | pobj : object of a preposition | The object of a preposition is the head of a noun phrase following the preposition, or the adverbs “here" and “there". (The preposition in turn may be modifying a noun, verb, etc.) Unlike the Penn Treebank, we here dene cases of VBG quasi-prepositions like “including", “concerning", etc. as instances of pobj. (The preposition can be called a FW for “pace", “versus", etc. It can also be called a CC { but we don't currently handle that and would need to distinguish from conjoined prepositions.) In the case of preposition stranding, the object can precede the preposition (e.g., “What does CPR stand for?").<br>I sat on the chair" pobj (on, chair) |
| **Rule 4** | xcomp: open clausal complement | An open clausal complement (xcomp) of a VP or an ADJP is a clausal complement without its own subject, whose reference is determined by an external subject. These complements are always non-finite. The name xcomp is borrowed from Lexical-Functional Grammar.<br>e.g. He says that you like to swim" xcomp(like, swim)<br>I am ready to leave" xcomp(ready, leave) |
| **Rule 5** | rcmod: relative clause modifier | A relative clause modifier of an NP is a relative clause modifying the NP. The relation points from the head noun of the NP to the head of the relative clause, normally a verb.<br>e.g. I saw the man you love rcmod(man, love)<br>I saw the book which you bought rcmod(book,bought) |

*Table 11 dependency Relations for Object Detection*

### 6.3.2. Object detection Algorithm.

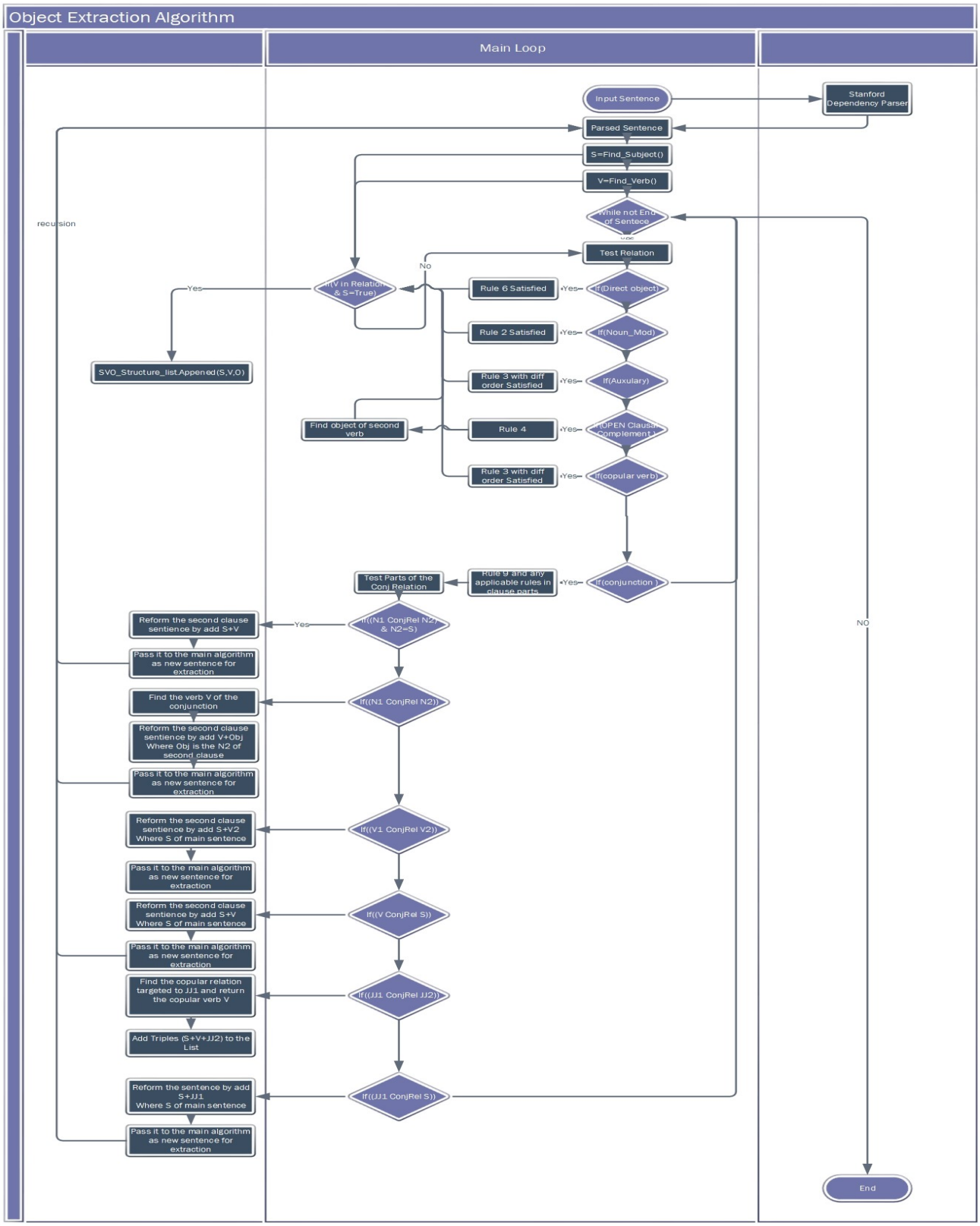


*Figure 16 Object Extraction Algorithm*

## 6.4. Adverbs Extraction.

An adverb is a word that modifies a verb, adjective, another adverb, determiner, noun phrase, clause, or sentence. It is typically express manner, place, time, frequency, degree or level of certainty. Adverbs can be realized by single words (adverbs) or by multi-word expressions (adverbial phrases and adverbial clauses). [82]

In our application of Hate speech and offensive language detection, adverbs play a significant rule in hate orientation and it express Hate sentiment level.

### 6.4.1. Adverb Manipulation Rules and Assumptions.

Rule 1. Adverbs works as modifiers of verb or verb phrases, it provides information about the manner, place, time, frequency or certainty. According to adverb definition above.

| Example | Adverb Type |
|---|---|
| **I worked Hardly** | Manner |
| **We study Her in polytechnic** | Place |
| **They are coming tomorrow** | Time |
| **I confidently done the project** | Certainty |

*Table 12 Rule 1 Example of Adverbs Types*

Rule 2. Adverbs works as a modifier of adjectives or other adverbs to express degree E.g.:

| Example | Adverb Type |
|---|---|
| **I am quite close to you** | Adverb **quite** modifies the adjective **close** |
| **I was running very happily** | Adverb very modify Adverb happily |

*Table 13 Rule 2 Example of Adverbs modify Adjective*

Rule 3. Adverbs works as a modifier to noun phrases, prepositional phrases, clauses or sentence, as in the following examples:

| Example | Adverb Type |
|---|---|
| **I eat only the banana on the table** | Adverb **only** modifies the noun phrases **the banana** |
| **They jumps almost to the swimming pool** | Adverb **almost** modify prepositional phrase **to the swimming pool** |
| **Certainly we have to study for exams** | Adverb **Certainly** modify sentence **we have to study for exams** |

*Table 14 Rule 3 Example of Adverbs modify noun phrase , prepositional phrase clauses or sentences*

Rule 4. This assumption assume that the Noun phrases works as a syntactical adverb modifier of the sentence, this assumption defined by Stanford typed dependency and it assumes that it occurs in the following places:

a. A measure phrase, which is the relation between the head of an ADJP/ADVP/PP and the head of a measure phrase modifying the ADJP/ADVP.
b. Noun phrases giving an extent inside a VP which are not object.
c. Financial constructions involving an adverbial or PP-like NP

d. floating reflexive.
e. certain other absolute NP constructions.

This assumption identified by Stanford Typed Dependency as an extended Adverbial phrase as in table [18].

| Rules | Dependency Relation | Description |
|---|---|---|
| **Rule 3** | **Advcl** adverbial clause modifier | An adverbial clause modifier of a VP or S is a clause modifying the verb (temporal clause, consequence, conditional clause, etc.).<br>e.g.The accident happened as the night was falling" advcl(happened, falling)<br>If you know who did it, you should tell the teacher" advcl(tell,know) |
| **Rule 1 ,2** | **Advmod** adverbial modifier | An adverbial modifier of a word is a (non-clausal) adverb or adverbial phrase (ADVP) that serves to modify the meaning of the word. e.g.<br>Genetically modified food" advmod(modified, genetically)<br>less often" advmod(often, less) |
| **Rule 4** | **Npadvmod** Noun phrase as adverbial modifier | This relation captures various places where something syntactically a noun phrase (NP) is used as an adverbial modifier in a sentence. These usages include: (i) a measure phrase, which is the relation between the head of an ADJP/ADVP/PP and the head of a measure phrase modifying the ADJP/ADVP; (ii) noun phrases giving an extent inside a VP which are not objects; (iii) financial constructions involving an adverbial or PP-like NP, notably the following construction $5 a share, where the second NP means“per share"; (iv) floating reflexives; and (v) certain other absolutive NP constructions. A temporal modifier (tmod) is a subclass of npadvmod which is distinguished as a separate relation. e.g.<br>The director is 65 years old" npadvmod(old, years)<br>feet long" npadvmod(long, feet)<br>Shares eased a fraction" npadvmod(eased, fraction)<br>IBM earned $ 5 a share" npadvmod($, share)<br>The silence is itself significant" npadvmod(significant, itself)<br>90% of Australians like him, the most of any country" npadvmod(like, most) |

*Table 15 Dependency Relations for Adverbs Detection*

### 6.4.2. Adverbs Detection Algorithm.

Adverbs usually associated by the verb once the verb have been detected the process of locating adverbs is started. adverb extraction algorithm is tested on the object detection algorithm loop for testing adverbial relation and add it to the adverb template structure list. Figure [17] show the flowchart of the algorithm.

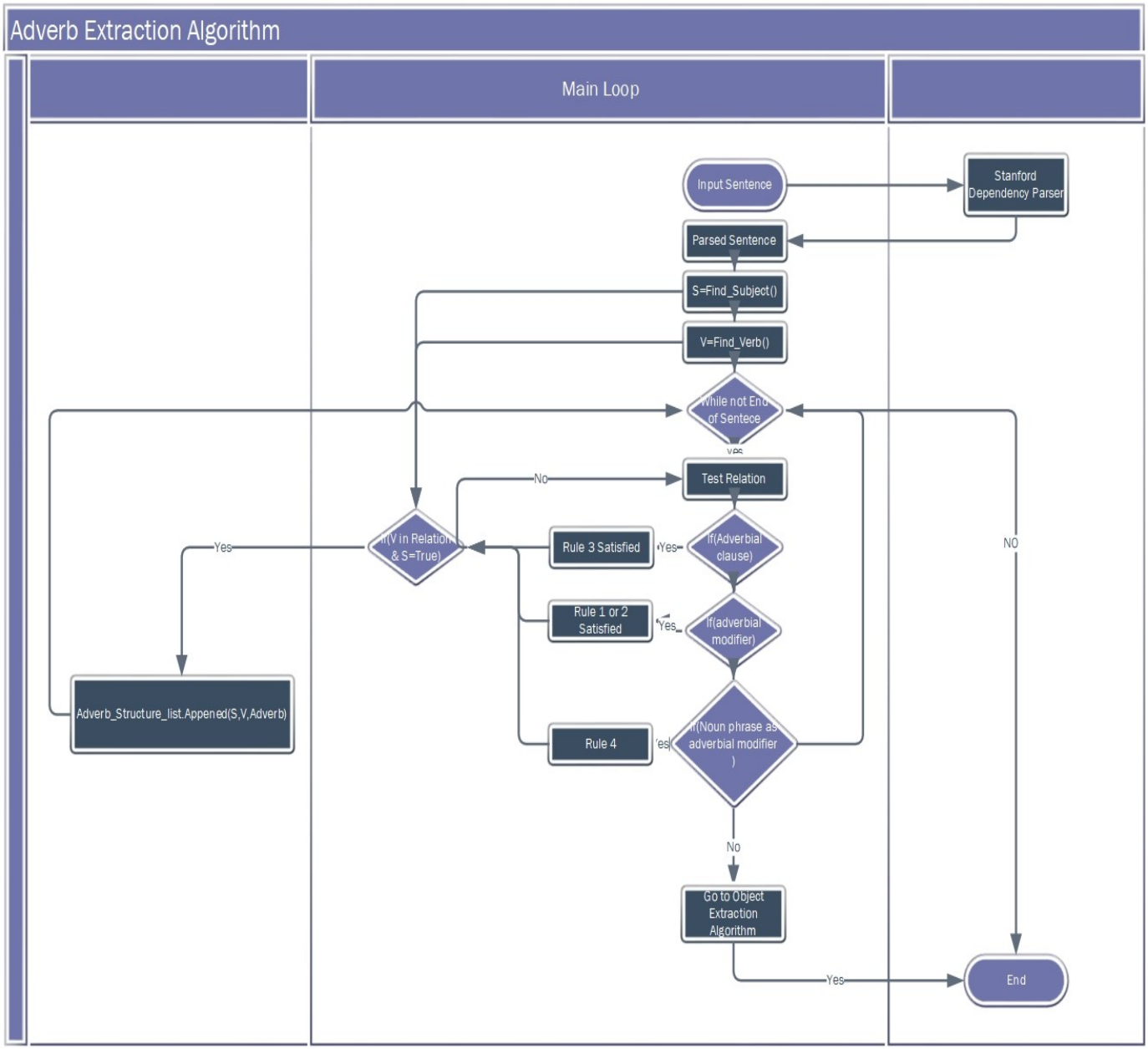



*Figure 17 Adverb Extraction*

## *6.5. Adjective Extraction.*

Adjective, in linguistics, is a describing word, its role is to qualify a noun or noun phrase, giving more information about the object signified. [83]

### 6.5.1. Adjective Manipulation Rules and Assumptions.

Rule 1. Adjective can be an Attributive Adjective are part of the noun phrase headed by the noun they modify as we see on consistency parse tree on analysis part. Attributed adjective can be presented before or after noun it modify e.g. “good students” or “students good”. It can be identified by testing the dependency relation (‘Amod’).

Rule 2. Adjective can be a Predicative Adjective that linked by copula or other linking parameters to the noun or pronoun they modify. as we have seen on copular dependency on Object extraction

e.g. I am confident, confident is a predicate adjective.

Rule 3. Adjective can work as a noun it called a Nominal Adjective. e.g. I read the sad book, sad is a nominal adjective.

The following table [19] show a dependency relation used for adjectives extraction:

| Rules | Dependency Relation | Description |
|---|---|---|
| **Rule 1** | **Amod** adjectival modifier | An adjectival modifier of an NP is any adjectival phrase that serves to modify the meaning of the NP. e.g. He eats red meat" amod(meat, red) |
| **Rule 3** | **Acomp** adjectival complement | An adjectival complement of a verb is an adjectival phrase which functions as the complement (like an object of the verb). e.g. "She looks very beautiful" acomp(looks, beautiful) |
| **Rule 2** | **Cop** copula | A copula is the relation between the complement of a copular verb and the copular verb. We described it before on object detection section and it usually came after auxiliary verbs works as an object. |

*Table 16 Dependency Relations for Adjectives Detection*

### 6.5.2. Adjective Detection Algorithm.

The process of adjectives phrases extraction is started when the subject and verb is not located in the sentences, in this case the clause is either a noun phrases or an adjective phrase the algorithms start examining the adjective relations and add it to the list of adjectives template structure list as in algorithm figure [18].

## *6.6. Noun Attributes Extraction.*

Nouns, in linguists described as words that refer to a person, place, thing, event, substance, quality, quantity, etc. [84] Noun can be used as subject or object or as descriptive that works as an attributes of other nouns in this section we will describe Noun used as an Attribute of other noun or noun phrases.

| Noun POS tag | Meaning |
|---|---|
| **NN** | Singular or mass e.g. book |
| **NNS** | Plural e.g. books |
| **NNP** | Proper Noun Singular e.g. Johan |
| **NNPS** | Proper Noun Plural e.g. Vikings |
| **PDT** | Predeterminer e.g. both the boys |
| **POS** | Possessive endings e.g. friend's |
| **PRP** | Personal Pronoun e.g. I he she |

*Table 17 Noun POS tags defined by penney treebank*

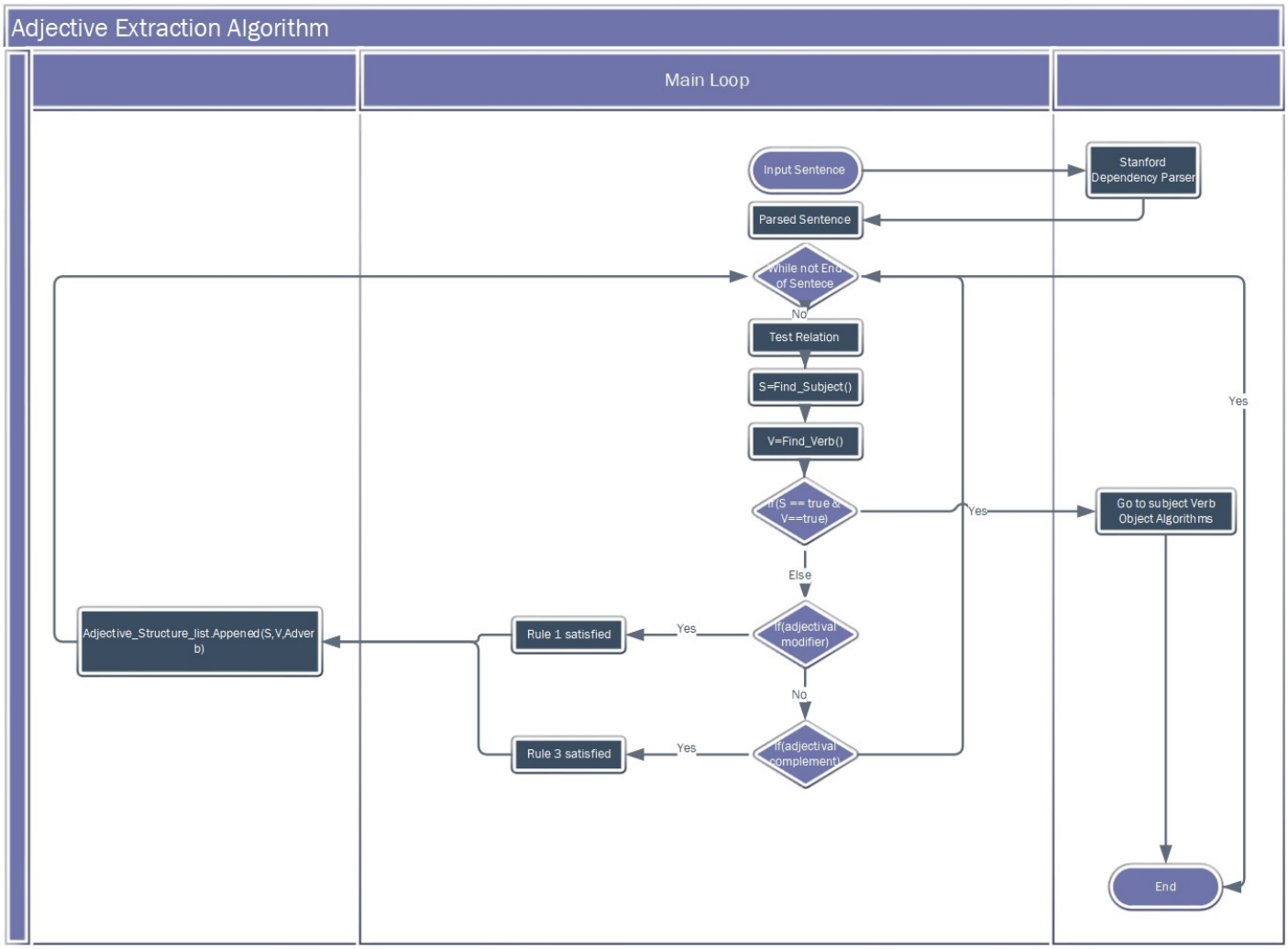



*Figure 18 Adjective Extraction Algorithm Noun Attributes*

### 6.6.1. Noun Attributes Detection Algorithm.

In the same way that adjectives have been detected, the algorithm starts when subject and verb is not located in the sentences, in this case the clause is either a noun phrases or an adjective phrase the algorithms start examining the Noun attributes and add it to the list of Noun template structure list as in figure [19]. The process of identify Noun attribute is by testing the noun POS defined by penny treebank as in table [20].

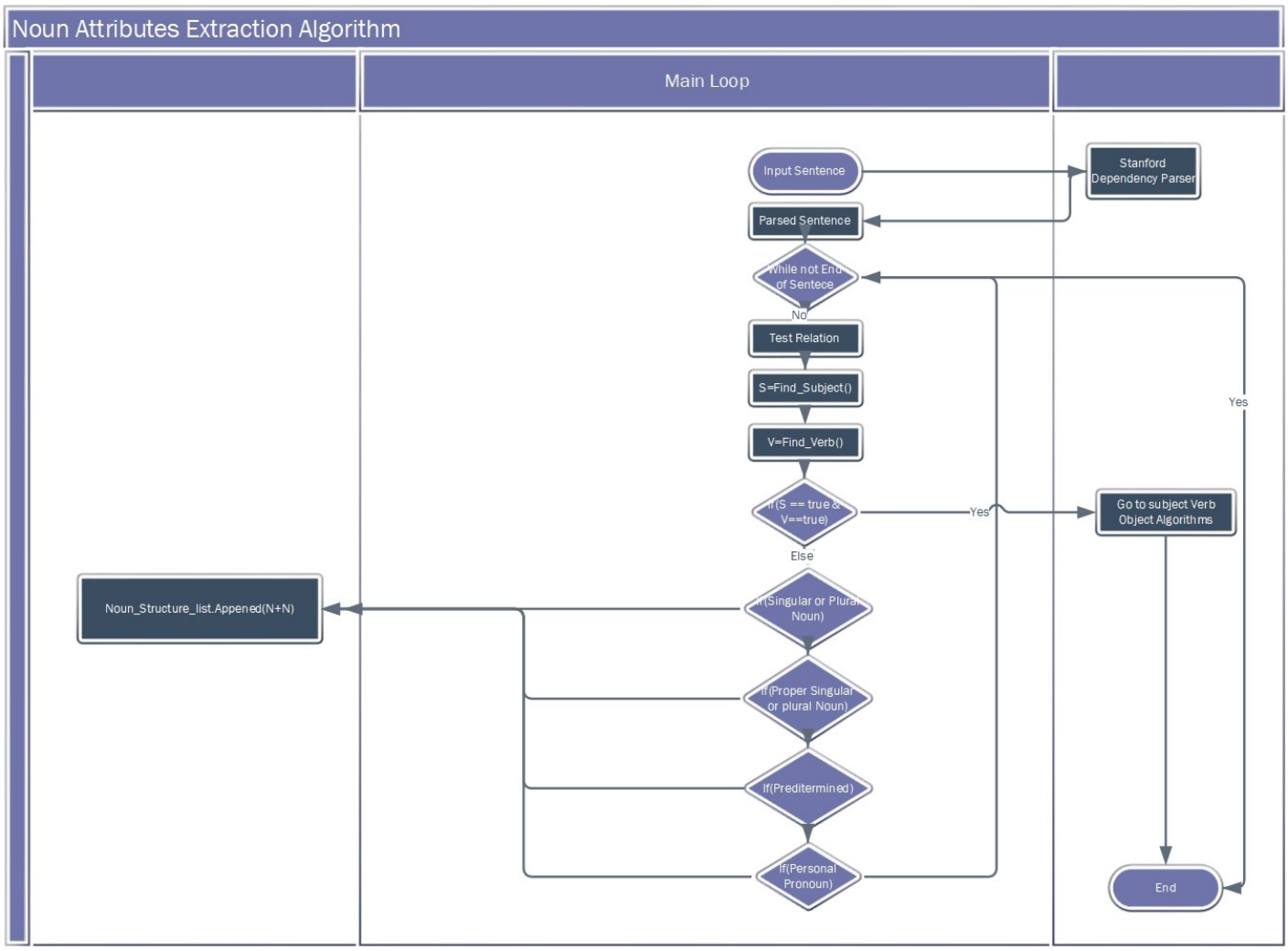



*Figure 19 Noun Attributes Detection*

# CHAPTER 7.

# (LINGUSTIC STRUCTURE FEATURE EXTRACTION SYSTEM DESIGN) (LSFES)

In this section we will describe the final system Design and then in the following sections we will describe its interface to machine learning system.

## 7.1. LSFES Block Diagram

Block diagram figure [20] simply show the general system component. The input text data collected by fetching tweets from twitter API using some patterns of hate vocabulary then it fetches tweets from user's profiles that contains hate and none hate tweets, or by sourcing a dataset. Preprocessing used to clean tweets and keep the original sentence structure –unlike preprocessing of machine learning the process of stemming and Stopword elimination are skipped to keep the original sentence structure. Stanford CoreNLP pipeline package is used throughout system development. The output of LSFES System is a set of Natural Language features used as an interface to machine learning system.

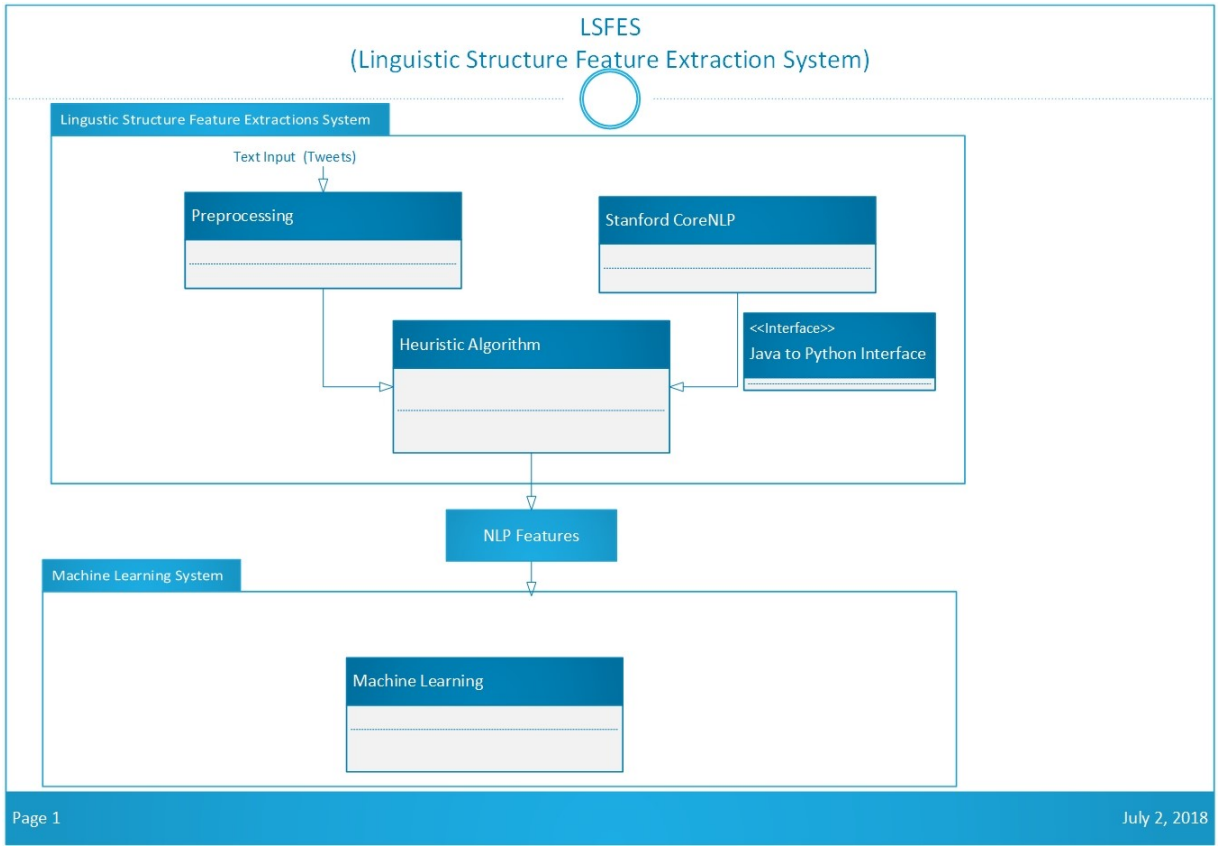



*Figure 20 Linguistic Structure Features Extraction Block Diagram*

## 7.2. LSFES System Design.

The Heuristics algorithms developed before integrated as a single unit that process the input sentence into syntactical structure by Stanford dependency parser and the main loop of the system create a heap of processing tree. this extensive manipulation tee result from recurrent nature of algorithm applied to each constituent of clauses and sentence that have been splitted in the first stage of sentence manipulation process.

We make use of Stanford dependency parser to transform an input sentence to its syntactical structure The parsed result consists of a set of directed syntactic relations between the words in the sentence. As we see in the algorithm figure [21] we first identify the clauses in the input sentence by extracting the subject relation and process forward to obtain the head word of all the constituents of each clause to gat clause type as ‘SV’ or ‘SVO’ if Verb have been detected the algorithm loop starting from verb index in the tree looking for the object by apply the object extraction algorithm. If subject relation is not detected the algorithm start looking for verb to extract the predicate clause in form of ‘VO’ or ‘VC’ followed by object or complement by apply object extraction algorithm. LSFES create an informative feature that doesn’t directly appear in the sentence. The feature constituents refer to a word or auxiliary verb that are placed anywhere within the sentence. For example, we replace the relative pronoun who or which by its antecedent, which is obtained by ‘rcmod’ relation.

In case of conjunction relation that link more sentences or clause we applied a set of conditional rules testing the conjunction parameters between linked sentences. A coordinated conjunction connects two or more parts of the sentence by coordinator as ‘and’ or ‘or’. CCs are detected by the Stanford parser and indicated by dependency relations such as (‘conj’). If a Conjunction is present in a constituent of a clause, the system replaces each of its conjoints and reformulate the sentences to its original form as it was before joined by conjunction and it passed to the algorithm for recurrent search as a new sentence for feature extraction.

Adverbs associated by the verb and the adverb extraction algorithm executed during the loop of object extraction by testing the adverbial relations. If clause haven’t detected by subject or verb relation, then the algorithm traverse dependency tree looking for adjective or noun attributes relations that express and modify the noun or adjective this feature rated as a most informative features that carrying the significant parts of the constituents of the clauses within the sentences and in our case of hate speech and offensive language the adjective and noun are the dominant components of speech that express the source sentiment by describing the target state. Figure [21] show LSFES workflow as an integration of algorithms designed in previous chapter.

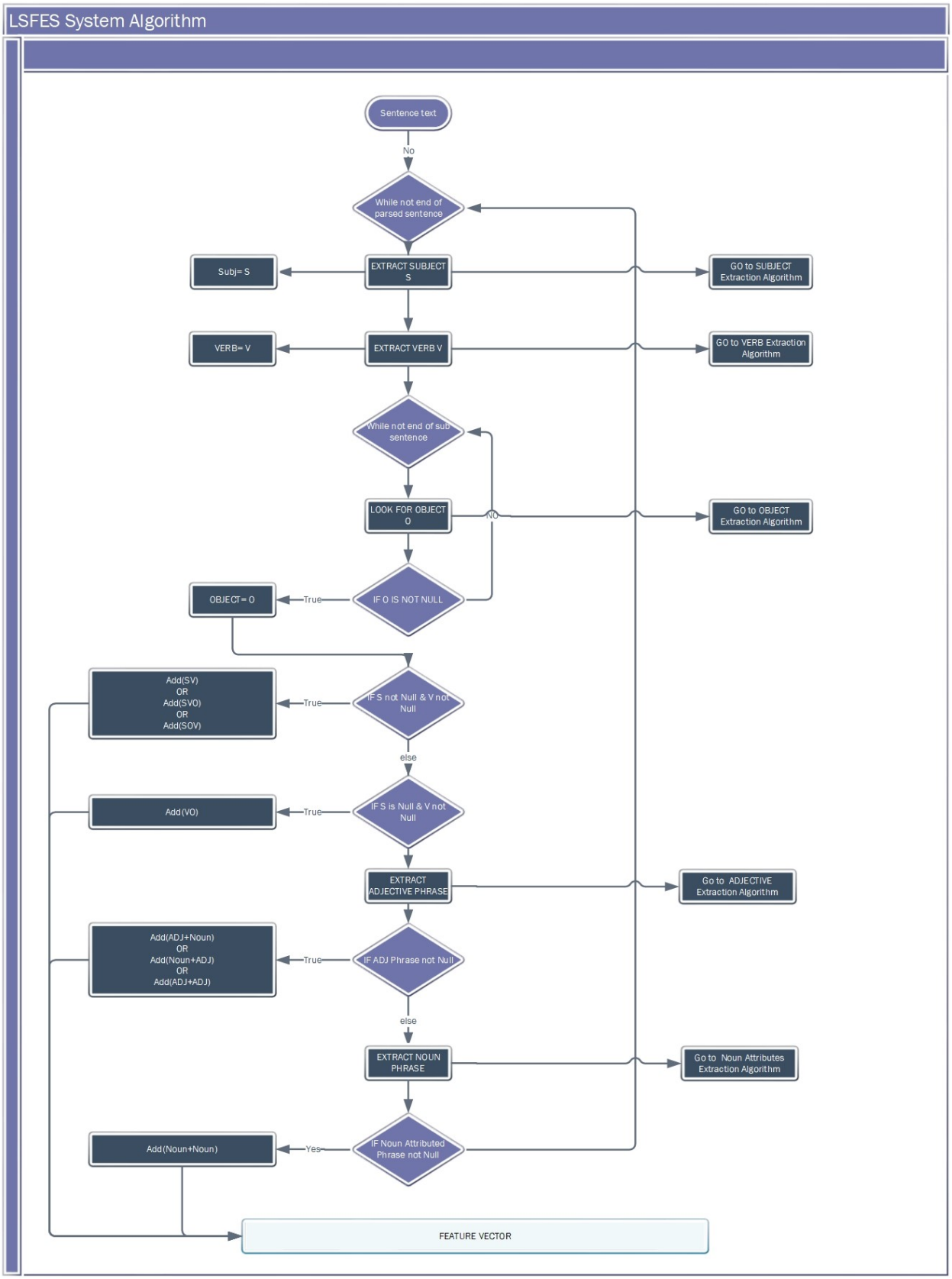



*Figure 21 LSFES Main Algorithms*

# CHAPTER 8.

# (LSFES RESULTS AND DISCUSSION).

This section describes the results by provide a case study example for analysis and then we described the syntax structure and how sentence components connected each other in relations to form its context, we showed how semantic network represented and how our system extracts this relation and prepare the features. we have presented a philosophical example to motivate our feature extraction preparation in ordered to improves learning performance. Then we discussed about future use of LSFES system including vectorizers, vocabulary extension, n-gram meaningful set generation and feature extraction.

## 8.1. LSFES Features Results and Case Study.

In this section we will describe the features output of the LSFES system, describing its interface to machine learning system. Then in the following section we will discuss and compare LSFES system with other system as Stanford OpenEI, ClauseIE and other tools.

As we discussed in analysis part how features generated based on structural analysis and its representation, in this section we will discuss its result by example and show how the NL sentence meaning converted to features sets. Let us bring the example described On analysis part we find how features represent the semantic network of sentence [Anna kills wild wolf and takes it away] and its philosophy. the dependency parse tree generated by Stanford Parser showed in figure [22] rooted on the first verb 'kills' by applying the rules and assumptions on previous sections traversing from the head down on the left for subject extraction 'Anna' is the nominal subject then iterating over the rest of tree looking for the object located on node 4 once conjunction has detected the sentence splitted and the subject linked to the forked sentence [Anna takes it away] as we discussed on Rule 9 of Object Extraction the process of extraction continue in recurrent way the last adjective phrase extracted is [wild wolf] that works as an object of the root verb takes.

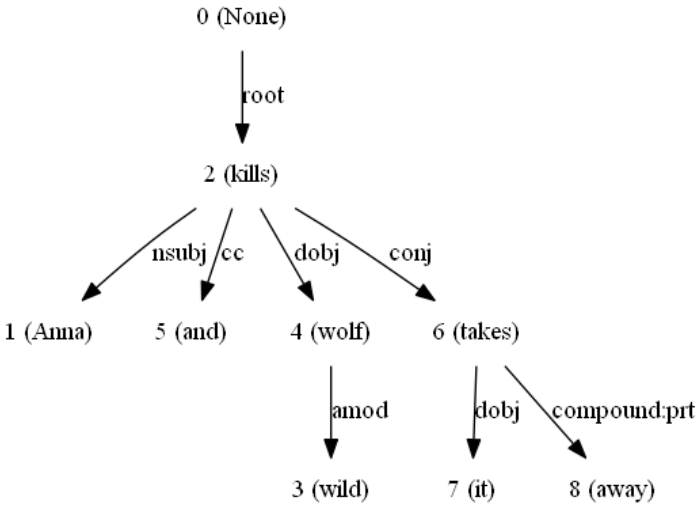



*Figure 22 Dependency parse tree of sentence [Anna kills wild wolf and takes it away ]*

So from sentence [Anna kills wild wolf and takes it away] The Features generated by LSFES are [ Anna kills, Anna kills wolf, Anna takes, Anna takes it, takes it, kills wolf, wild wolf]

figure (23) show word network of features, the developed technique extracts each meaningful phrases from the sentence as 1-3 ngrams generating all possible new meaningful phrases by analyzing sentence grammatical structure resulting on a set of n-grams set we note that the meaning concentrated on some words and those words can show a larger relational network among others, those affected words contribute more in formation of the main concept of the sentence and can generate more new meaningful phrases by the algorithm, as an example :

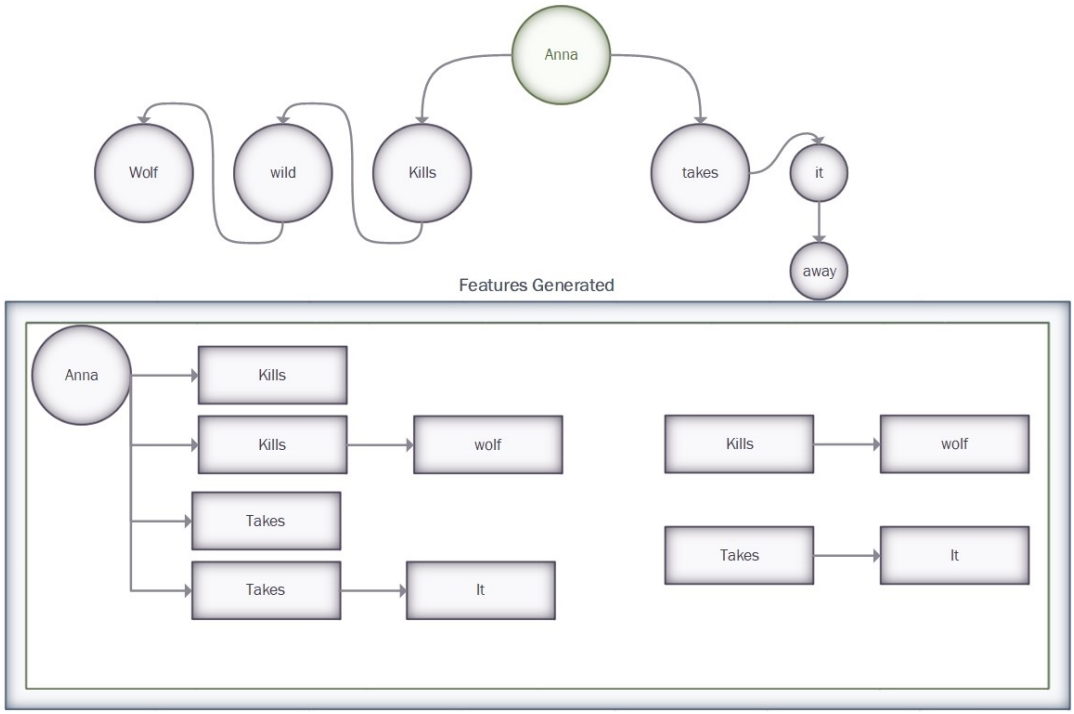



*Figure 23 Features Generated by linguistic structure features extraction technique of sentence : [Anna kills wild wolf and takes it away]*

as we see in figure (23), word “Anna” as a target generated four set of phrases as [Anna kills, Anna kills wolf, Anna takes, Anna takes it] this more relational net can contribute on the main concept of the sentence that can be “Anna is champion”. As a consequence, word “kills” generate three meaningful phrases as features [Anna kills, Anna kills wolf, kills wolf] those relations show that the “kill” is an important event and continued more in Annas champions. Although those words have occurred only one time in the original sentence, we tried to develop a Linguistic Structure Extraction Algorithm technique capable of extracting the hidden meaning by generating a maximum possible meaningful phrases as features set to be fit to machine learning system.

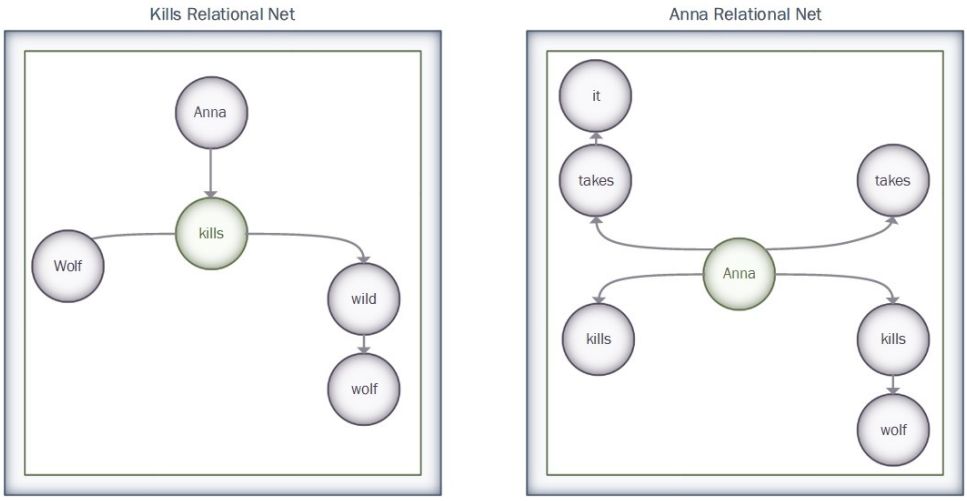



*Figure 24 words "Kills" and "Anna" Relational Network of sentence : [Anna kills wild wolf and takes it away]*

We have seen on feature engineering and dimensionality reduction how feature represented in a numerical vector form and how words weighted by the selected feature extraction model and vectorization as TF-IDF or Count vectorization that weighting the words based on their frequencies of occurrence. Therefore, the novel behind generating a redundant phrases of significant words as features was to increase its weight on transformation phase so that high level feature extraction technique aimed to transform the high level sentence meaning to Machine Level understanding.

In analogy of Human level learning, if we try to teach children a sentence to be understood, we try formulating the sentence in different phrases of shorter length concentrated on the significant terms, as an example if we teach children the sentence [Anna kills wild wolf and takes it away] we will try formulating the sentence in different form like “Anna kills wolf!”,” Anna kills”, ”Anna takes it Away”, “Anna takes it” and so on, this facility make it understood easily. We can imagine this conversation:

**Teacher**: Hi! [Anna kills wild wolf and takes it away]

***Children****: What!?*

***Teacher****: hmm! [Anna kills]*

***Children****: whom killed!??*

***Teacher****: [Anna kills wolf]*

***Children****: oh! why!?*

***Teacher****: [Anna kills wild wolf]*

***Children****: I understand now, then?*

***Teacher****: [Anna takes it]*

***Children****: where?*

***Teacher****: [Anna takes it Away]*

***Children****: Ok I understand now [Anna is champion] and [Anna hate wolf]*

***Teacher**: bravo, perfect.*

This philosophy of redundant expression of sentence was the base of our algorithm design, in which heuristic algorithm extract the hidden meaning and terms importance in measures of high word network length and number of possible relations linked to it, this can be implemented by deep analysis of linguistic grammatical structure and clauses coherent and extracting all possible phrases as [Noun phrases, adjective phrase, adverb phrase and all forms of verbal phrases] and in additions the links between sentences and clauses by conjunctions as we describe on analysis stage.

On the machine level, the result is a numerical vector such that the magnitude of important terms or phrases have incremented in correspondence to its importance and this is one of the target of this linguistic structure features preparation and extraction technique.in the following table a set of features generated by the LSFES system

| Sentence | LSFES Features |
|---|---|
| **"The baby is cute "** | [ 'baby cute', 'baby is cute'] |
| **"bill is an honest man"** | ['bill man', 'bill is honest', 'honest man'] |
| **"Clinton defeated Dole"** | [ 'Clinton Dole', 'defeated Dole'] |
| **"Dole was defeated by Clinton"** | [ 'Dole defeated', 'Dole was defeated', 'Clinton defeated Dole', 'was defeated', 'Clinton defeated'] |
| **"Tom likes to eat fish"** | [ 'Tom likes' , 'Tom likes eat' , 'Tom eat fish' , 'likes eat' , 'eat fish' ] |
| **"what she said"** | [ 'said what' , 'she said' ] |
| **"What she said makes sense"** | [ 'said What', 'she said', 'makes sense'] |
| **"That she lied was suspected by everyone"** | [ 'That suspected', 'That was suspected', 'everyone suspected That', 'she lied', 'was suspected', 'everyone suspected' ] |
| **"I saw the man you love"** | [ 'I saw', 'I saw man', 'saw man', 'you love'] |
| **"Sara fed the dogs"** | [ 'Sara fed' , 'Sara fed dogs' , 'fed dogs' ] |
| **"She sent him a present"** | [ 'She sent', 'She sent him', 'She sent present', 'sent him', 'sent present' ] |
| **"She is waiting for Tommy"** | [ 'She waiting', 'She is waiting', 'is waiting' ] |
| **"Reagan has died"** | [ 'Reagan died' , 'Reagan has died' , 'has died' ] |
| **"Kennedy has been killed"** | [ 'Kennedy killed' , 'Kennedy has killed' , 'Kennedy been killed' , 'has killed' , 'been killed' ] |
| **"Bill is big and honest"** | [ 'Bill big' , 'Bill is big' , 'Bill is honest' ] |
| **"What does CPR stand for"** | [ 'stand What' , 'does stand' , 'CPR stand' ] |
| **"He says that you like to swim"** | [ 'He says' , 'like swim' , 'you like' , 'you like swim' ] |
| **"If you know who did it, you should tell the teacher"** | [ 'should tell' ,' tell teacher' , 'you know' , 'who did' , 'who did it' , 'did it' , 'you tell' , 'you should tell' , 'you tell teacher' , 'tell know' ] |
| **"Genetically modified food"** | [ 'Genetically food' ] |
| **"The silence is itself significant"** | [ 'silence significant' , 'silence is significant' , 'itself significant' ] |
| **"He eats red meat"** | [ 'He eats' , 'He eats meat' , 'eats meat' , 'red meat' ] |
| **"She looks very beautiful"** | [ 'She looks' , 'She looks beautiful' , 'looks beautiful' , 'She looks very' ,' looks very' ] |
| **"I eat only the banana on the table"** | [ 'I eat', 'I eat banana', 'eat banana', 'I eat only', 'eat only'] |

*Table 18 LSFES FEATURES GENERATION EXAMPLES*

## *8.2. LSFES vs Open Information Extraction (OIE) tools.*

Since we had no access to annotated data, LSFES as with different approaches including OpenIE (Open Information Extraction), ClauseIE working with non-labeled data those approaches use linguistic

information from the text, among other techniques and algorithms for attempting to extract the relations without the need of labelled data for a trained model.

Stanford's OpenIE [16] is the first of these tools it works by utilizing two classifiers, both applied on linguistic information from the text. First classifier works at the text level and attempts to predict self-contained sentences from the text. Once these sub-sentences are extracted, the second classifier use its linguistic structure to identify the relation triples by traversing on dependency tree and select the candidate arcs. AllenAI's OpenIE [17] process the linguistic text by handling consistency parse tree to extract the POS and NP-chunks of the sentence parsed by Apache OpenNLP parser, and then apply a regular expression to look up the pattern relations by searches for clauses in the format V | VP | VW*P, where V is a verb or adverb, W is a noun, adjective, adverb, pronoun or determiner, and P is a preposition, particle or information marker. after clause identification, it uses a custom classifier called ARGLEARNER for arguments extraction Arg1 and Arg2.

The second interesting tool is ClausIE, a Clause-Based Open Information Extraction [18] from the Max-Planck-Institute use a dependency parser for parsing sentence and use rules to find relations.

ClauseIE starts by first finding clauses (candidate relations) by searching for subject dependencies and then parse the entire sentence to get the contents of this relation. This tool concentrated on the verbal phrases as a clause type as in table [22].

| | Pattern | Clause type | Example | Derived clauses |
|---|---|---|---|---|
| | | | **Basic patterns** | |
| $S_1$: | $SV_i$ | SV | AE died. | (AE, died) |
| $S_2$: | $SV_eA$ | SVA | AE remained in Princeton. | (AE, remained, in Princeton) |
| $S_3$: | $SV_cC$ | SVC | AE is smart. | (AE, is, smart) |
| $S_4$: | $SV_{mt}O$ | SVO | AE has won the Nobel Prize. | (AE, has won, the Nobel Prize) |
| $S_5$: | $SV_{dt}O_iO$ | SVOO | RSAS gave AE the Nobel Prize. | (RSAS, gave, AE, the Nobel Prize) |
| $S_6$: | $SV_{ct}OA$ | SVOA | The doorman showed AE to his office. | (The doorman, showed, AE, to his office) |
| $S_7$: | $SV_{ct}OC$ | SVOC | AE declared the meeting open. | (AE, declared, the meeting, open) |
| | | | **Some extended patterns** | |
| $S_8$: | $SV_iAA$ | SV | AE died in Princeton in 1955. | (AE, died)<br>(AE, died, in Princeton)<br>(AE, died, in 1955)<br>(AE, died, in Princeton, in 1955) |
| $S_9$: | $SV_eAA$ | SVA | AE remained in Princeton until his death. | (AE, remained, in Princeton)<br>(AE, remained, in Princeton, until his death) |
| $S_{10}$: | $SV_cCA$ | SVC | AE is a scientist of the 20th century. | (AE, is, a scientist)<br>(AE, is, a scientist, of the 20th century) |
| $S_{11}$: | $SV_{mt}OA$ | SVO | AE has won the Nobel Prize in 1921. | (AE, has won, the Nobel Prize)<br>(AE, has won, the Nobel Prize, in 1921) |
| $S_{12}$: | $ASV_{mt}O$ | SVO | In 1921, AE has won the Nobel Prize. | (AE, has won, the Nobel Prize)<br>(AE, has won, the Nobel Prize, in 1921) |

S: Subject, V: Verb, C: Complement, O: Direct object, $O_i$: Indirect object, A: Adverbial, $V_i$: Intransitive verb, $V_c$: Copular verb, $V_c$: Extended-copular verb, $V_{mt}$: Monotransitive verb, $V_{dt}$: Ditransitive verb, $V_{ct}$: Complex-transitive verb

*Table 19 Patterns and Clause Types based on ('Randolph Quirk, Sidney Greenbaum, Georey Leech, and Jan Svartvik. A Comprehensive Grammar of the English Language. Longman, 1985.)*

Our tool, the so called LSFES use the dependency parse tree as clauseIE for Natural Language feature extraction that includes all meaningful constituents that can contribute on the final concept formulation of the sentence to be used as a Machine Learning Natural Language classification and detection. In the following table [23] a comparison between our tool, openIE and clauseIE in terms of use and functionality.

| OIE Tool | Usage | Analysis Based on | Pattern and clause type | Example | Extraction Output |
|---|---|---|---|---|---|
| **OpenIE** | • Information Extraction<br>• Question Answering<br>• Information Retrieval | • Dependency and Techniques<br>• AllenAI's OpenIE also based on Consistency | As in table [15] all verbal clauses | We stress that our method improves a supervised baseline. | **improves (our method, supervised baseline)** |
| **ClauseIE** | • Information Extraction<br>• Question Answering<br>• Information Retrieval | • Dependency<br>• Rule based Techniques | As in table [15] all verbal clauses | We stress that our method improves a supervised baseline. | **[(We, stress, that our method), (our method, improves , supervised baseline )** |
| **LSFES** | • NL Feature Extraction<br>• Information Extraction<br>• Question Answering<br>• Information Retrieval<br>• Ngram custom Vectorization technique.<br>• Vocabulary/Features augmentation. | • Dependency.<br>• Rule based Heuristics algorithms | Features as [(SV, SVO, SOV, VO, VOO, SVOO, SVA, VA),<br>(NN, NP),<br>(JJ JJ, N JJ,jj N, ADJP)<br>] | We stress that our method improves a supervised baseline. | **[ We stress , method improves baseline, improves baseline , method improves, supervised baseline ]** |

*Table 20 OIE systems comparisons.*

LSFES tool development follow the same path of ClauseIE tool, it is based on rules applied to dependency parse tree, as a contrast of ClauseIE we make triple extraction more concise with a maximum of three words per relation rather than using complete phrases in parties of the relation, as an example the sentence parsed dependency shown in figure [25], ClausIE extracts the following triples:

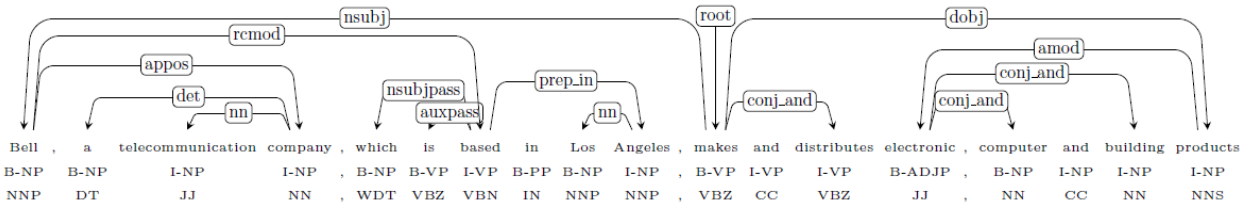



*Figure 25 An example sentence with dependency parse, chunks, and POS tags (chunks by Apache OpenNLP)*

[("Bell", "is", "a telecommunication company"),("Bell", "is based", "in Los Angeles"),

("Bell", "makes", "electronic products"),("Bell", "distributes", "electronic products"),

("Bell", "distributes", "computer products"),("Bell", "distributes", "building products").]

In witch subjects, verbs, objects or complements can be a clause phrases as "a telecommunication company", "is based", "in Los Angeles", in our LSFES tool rather it extracts relations as:

[Bell makes products, Bell makes, makes products, Bell distributes products, …] among others features the aim as we described in previous section was to extract all possible meaningful triples to be used for training

purpose so it was beneficial to scale down the length of triples fit to machine learning system to get a higher probability of matching on testing data.

## 8.3. LSFES Future Uses

LSFES tool can be used with different applications as Natural Language ML Features Extraction, Information Extraction, Information Retrieval and questions answering, it can be used as a built in tool embedded with in vectorizer for n-grams extraction and TFIDF support as we have seen how this tool extract the semantic network of sentences in redundant features to show words importance that will have impact on TFIDF Performance. It can be used as a vocabulary or features expending and augmentation.

### 8.3.1. NL Feature Extraction.

As we described in section [3.1] LSFES system used for feature generation from NL and perform an Interface to Machine Learning systems.

### 8.3.2. Information Retrieval/extraction and question Answering.

As OIE systems OpenIE and ClauseIE the system LSFES can be modified to retrieve and extract useful information by adding some rules applied to dependency parsed tree so that Subject, Object or prepositions can be an independent clause so that the extracted relation triples will be more descriptive. The grouping works as follows: if we have a tree with depth 4 headed at node (Verb) then the grouping works with 1st level children and a new tree is formed only with head node and its children so that nodes at level 2,3 or 4 collapsed under its parented node at level 1 as in figure [26].

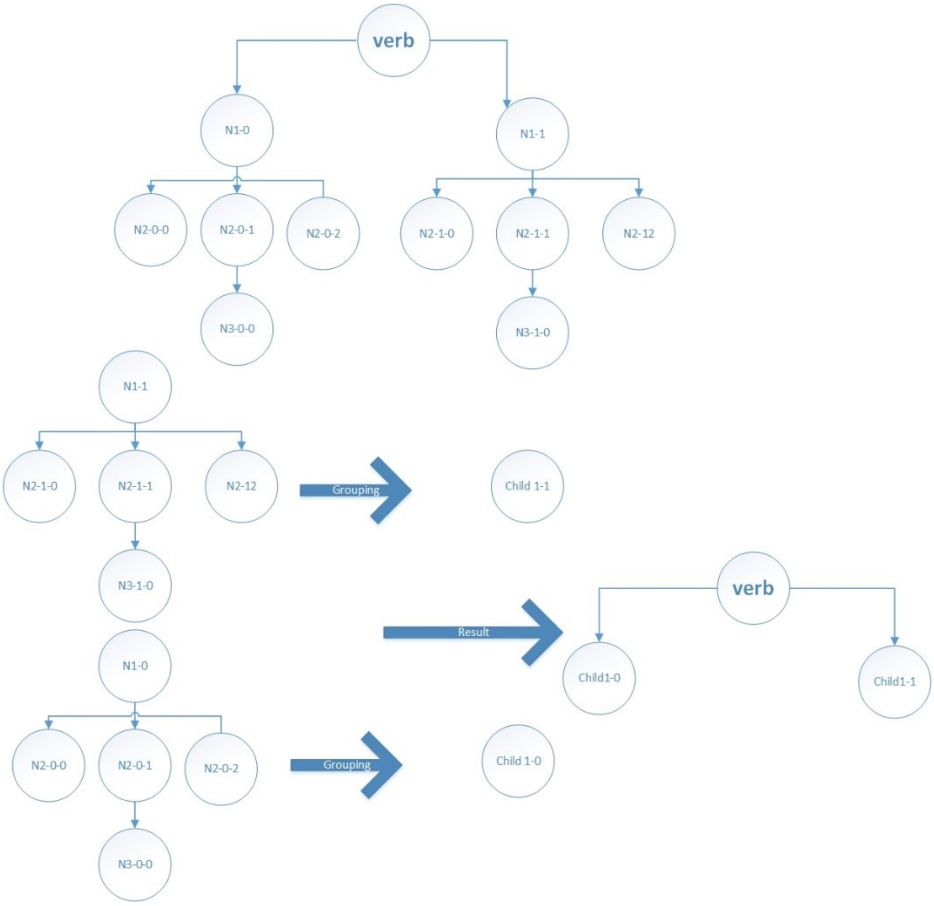



*Figure 26 Sentence Grouping Simplification*

### 8.3.3. Meaningful N-gram Generation.

LSFES tool can be used to generate a meaningful pairs or triples as an N-gram that is embedded within the vectorizer as TFIDFVectorizer or CountVectorizer, those vectorizers first generate an n-gram from text and then computes its terms frequency corresponding to its importance and representing it in a numerical vector to be fit to ML classifiers. The actual n-gram generation is done in a lazy method so that neighbors 1-n words grouped to gathers in this way too many un-meaningful n-grams is useless and has a direct impact on machine learning classifiers training performance as an example the sentence we described before [Anna kills wild wolf and takes it away] the vectorizer generate the following n-gram if n=3 then n-gram is:

[[Anna, kills, wild, wolf, and, takes, it, away] ,[Anna kills, kills wild, wild wolf, wolf and, and takes, takes it, it away], [Anna kills wild, kills wild wolf, wild wolf and, wolf and takes, and takes it, takes it away ]]

In this example many pairs, triples are considered as a useless for training as [‘it’, ’and’, ’away’, ‘wolf and’, ’and takes’, ’it away’, ’wild wolf and’, ’wolf and takes’] and so on these n-grams has an influence on training the models. Although, some techniques used in dimensionality reduction as filtration and word frequency method we described in dimensionality reduction chapter used to eliminate terms that has a frequency less than specific threshold and reduce the dimensionality of VSM (vector space model), but this doesn’t solve the problem and many of generated n-gram features frequently occurs in the dataset and it is uncorrelated to the sentence label and in most case doesn’t has a meaning. The problem will be increased with complex problems as in the sentence ['Bell , a telecommunication company , which is based in Los Angeles , makes and distributes electronic , computer and building products .'] in witch relations between words doesn’t occurs in sequence and its coherence along distances as in figure[27] bellow :

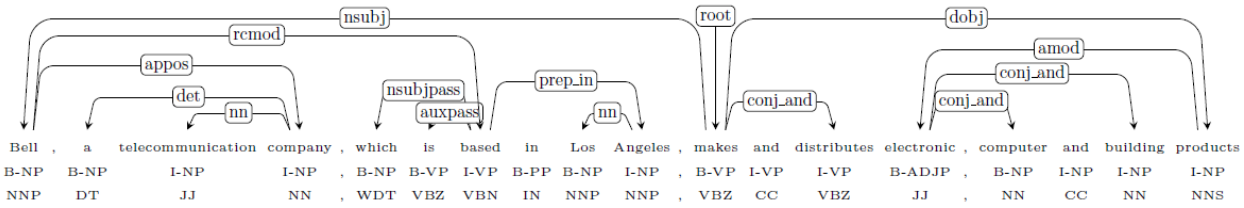



*Figure 27 An example sentence with dependency parse, chunks, and POS tags (chunks by Apache OpenNLP)*

So the n-gram generation will never represent the sentence meaning or getting any useful pairs or triples that can be efficiently fit to classifier for training. So our system LSFES use the Linguistic structure, grammatical relationships and word to word dependency for features inference that has a useful meaning and generate a redundant triples based on root word importance that has more relationships with other words within the sentence to improve the ML training performance. So the proposal of using this system within the implementation of the vectorizer for efficient n-gram semantic feature extraction figure [28].

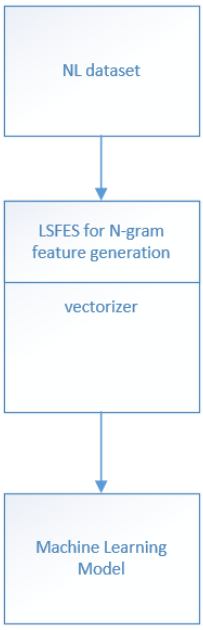



*Figure 28 N-Gram Feature Generation by Embedded LSFES & Vectorizer*

### 8.3.4. Features / Vocabulary Expanding.

LSFES tool by its development nature, try to extract all possible relations of a sentence corresponding to the meaning and node importance of the dependency tree. According to philosophy we propose in chapter [7] of teacher and children conversation, we have seen how redundant expression of sentence improve understanding this phenomenon can be modeled to machine learning system by feature pruning and expanding we propose a system as in figure [29] so that semantic features of sentences extracted the probability of correlation between individual features and the main concept of sentence (label) increased.

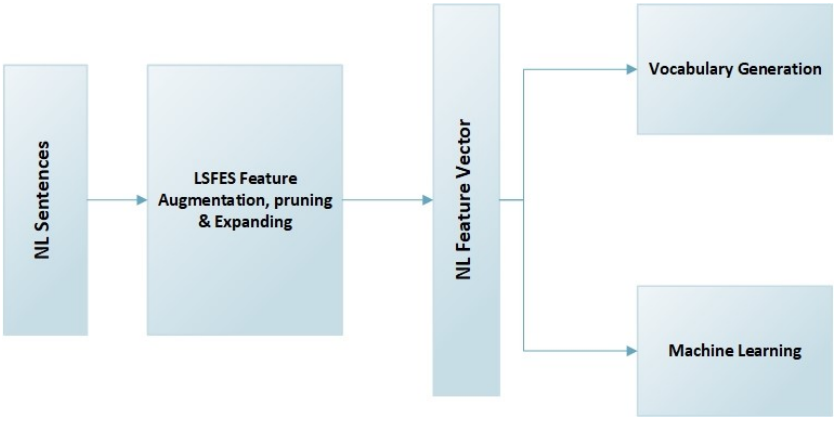



*Figure 29 LSFES for Feature/Vocabulary Expanding*

# III. THIRD PART: (LEARNING BASED SYSTEM).

This part present the machine learning system stages and components needed for hate speech classification we presented the system in chapters [9-13] including the dataset used, classification algorithms, feature extraction methods and the workflow we set for hate speech and offensive language classification.

# CHAPTER 9.

# (MACHINE LEARNING MODELS ANALYSIS).

Machine learning is a subfield of computer science, that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. [38]

A core objective of a learner is to generalize from its experience. Generalization in this context is the ability of a learning machine to perform accurately on new, unseen data after having experienced a learning data set. Mathematically, the training set is a statistical probability distribution, general unknown data that is represented later as the space of occurrences and the machine learning model has to build a general model about this space that enables it to predict in new sets that fit in this space.

From this view point, we can go through analysis of our machine learning algorithms that we used in our built models to know how feature extraction and the new technique we developed (Natural Language Structure feature extraction) fit in to several classifiers by proofing its individual mathematical assumptions.

**Machine learning Algorithms.**

In this section, we will take a tour through a selection of popular and powerful machine learning algorithms that are commonly used in our work. distinguishing differences among several supervised learning algorithms, exploring their strengths and weaknesses.

We will Introduce the concepts and engineering of selection of the appropriate classifiers for our model and features preparations.

Although Choosing an appropriate classification algorithm for a particular problem task requires practice, each algorithm has its own tricks and has its own mathematical assumptions.

In practice, there is certain classifier works best for all possible scenarios. so, it is always recommended that we compare the performance of a set of different learning algorithms to select the best model for the particular problem, so we have selected a set of algorithms to be tested in our model to get the best performance, the selection differ based on the number of features or samples, the amount of noise in a dataset, and whether the classes are linearly separable or not.

Machine learning tasks are typically classified into three broad categories, depending on the nature of the learning "signal" or "feedback" available to a learning system as we see in figure [30].

We will discuss mathematically each class of them focusing on the Supervised Algorithms used in our work to understand how the feature fit into the machine learning classifiers and the methodology of prediction.

This Engineering can give us an insight of selection of the appropriate algorithm to be fit by our data class.

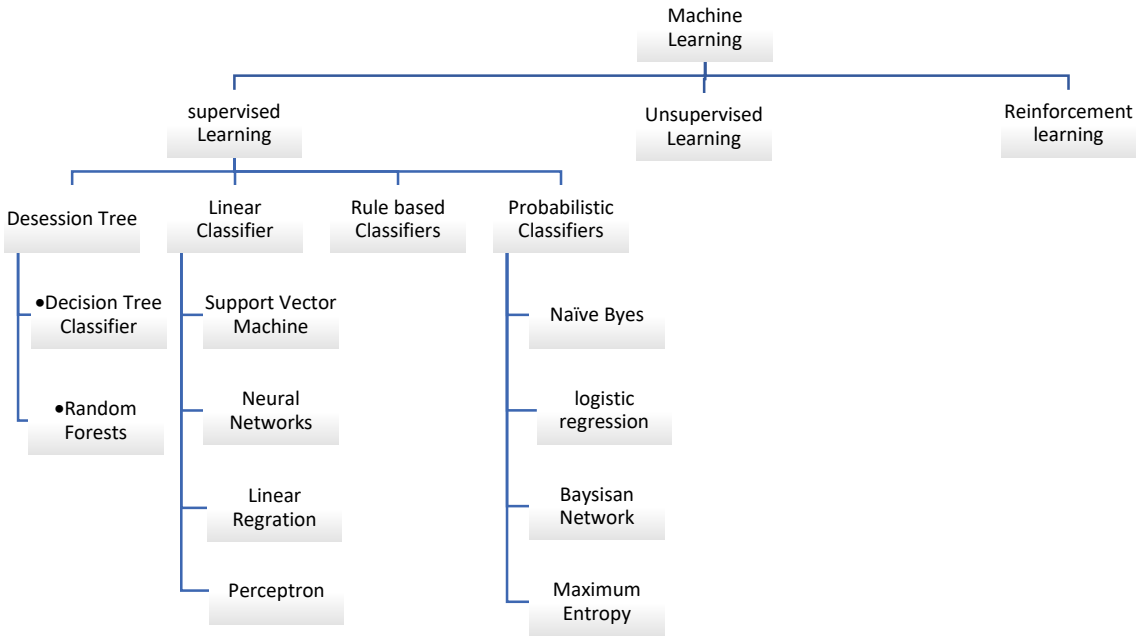



*Figure 30 Machine Learning Approach*

## 9.1. Supervised Learning

Supervised learning is the machine learning task of inferring a function from labeled training data. [37] In supervised learning, each sample is a pair consisting of an input object (typically a vector) and a desired output value, the supervisory signal. A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new samples on the base of the concept of generalization of the training space as in figure [31].In this section, we are going to analyze supervised algorithms used in our work as:

- Linear Classifier
  - Support Vector Machine.
    1. Linear SVC
    2. Non Linear SVC
  - Linear Regression
  - Perceptron
- Probabilistic Classifiers
  - Naïve byes
    1. Gaussian NB
    2. Multinomial NB
  - LogisticRegression
- Decision Tree Classifier
  - Random Forest Classifier
  - SGD Classifier

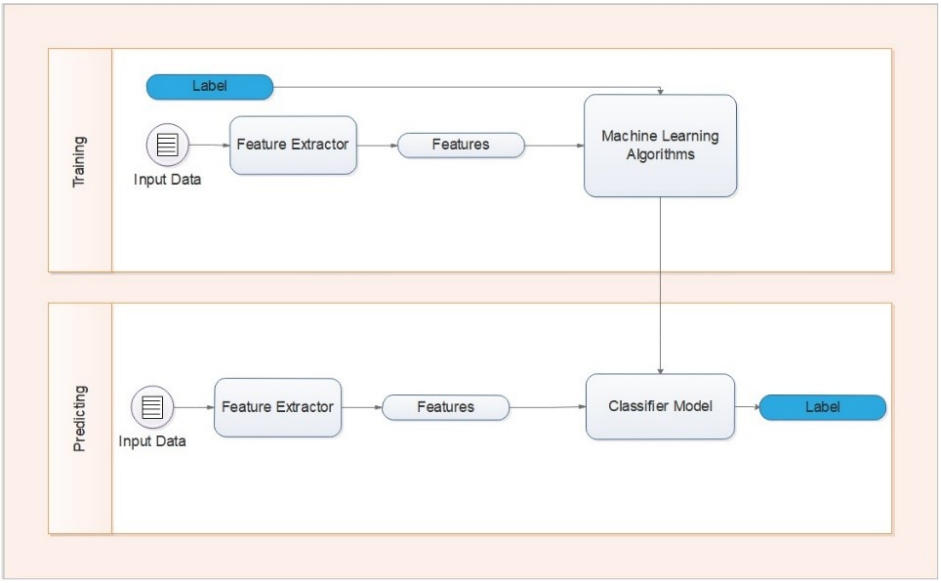



*Figure 31 Supervised Machine Learning*

### 9.1.1. Linear Classifiers

#### *9.1.1.1. Support Vector Machine.*

support vector machines are supervised learning models with learning algorithms that used for classification and regression analysis.

As an example of binary classification given a set of training samples, each sample belongs to one of two categories, SVM training algorithm builds a model that maps new samples into either class. SVM represent the samples as points in space separated by a hyperplane that set as wide as possible.

The SVM has an objective of hyperplane separation is to minimize the distance between the training data and the hyperplane so that the optimal hyperplane line achieved by the hyperplane that has the largest distance to the nearest training-data point, so that generalization error limits to minimum.

SVMs can perform a non-linear classification using the kernel trick, by implicitly representing the samples points into high-dimensional feature space.

SVM can be configured so that it works with unsupervised machine learning using clustering method embedded by SVM called SVC (Support Vector Clustering).

In some cases, the problem may be fit in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. So, it is efficiently to map the original finite-dimensional space into a higher-dimensional space, to make the hyperplane separation easier in that space. To facilitate the computation, SVM designed so that the dot products computed easily in terms of the variables in the original space, by defining them in terms of a kernel function $k(x, y)$ selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters $\alpha_i$ of images of feature vectors $x_i$ that occur in the data base. With this choice of a hyperplane figure [32]

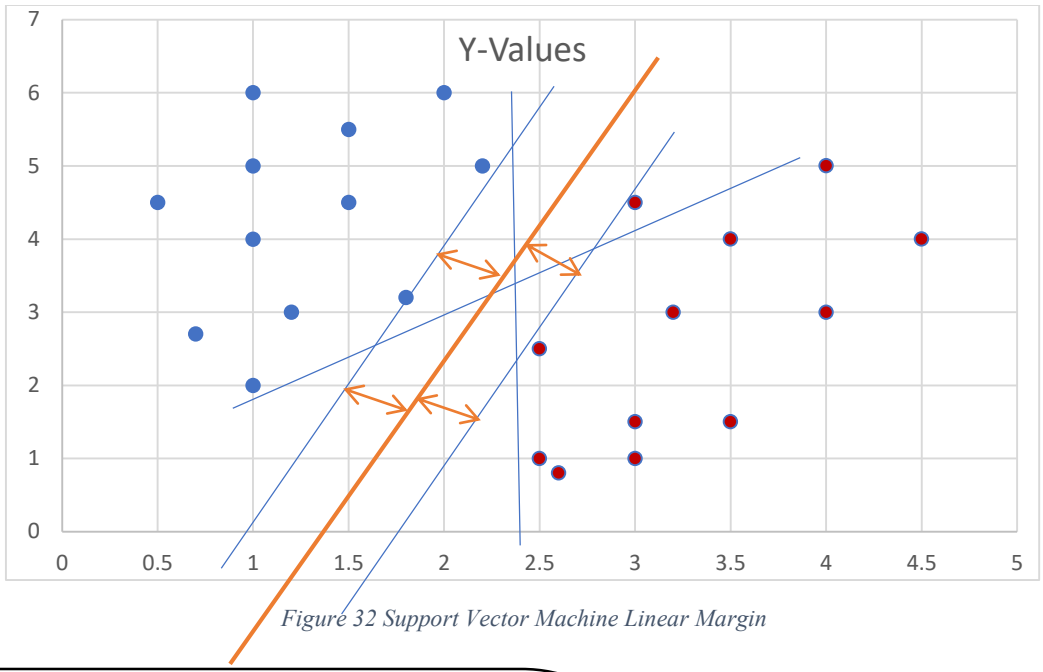



*Figure 32 Support Vector Machine Linear Margin*

An algebraic representation of the Linear SVM as follow:

*given the data*:

*The points* $(X_i, Y_i)$ *where* $i \subseteq \mathrm{N}, X_i \subseteq R^d, Y_i \subseteq \{1, -1\}$

*The hyperplane equation*: $W \bullet X + b = 0$

*All hyperplanes in* $R^d$ *are parameterize by a vector* $(W)$ *and a constant* $b$

*The Objective* :

find such a hyperplane $f(x) = sign(w \bullet x + b)$, that classify our data correctly.

Define the hyperplane $H$ as in figure [33] such that:
$xi \bullet w + b \geq +1$ *when* $yi = +1$
$xi \bullet w + b \leq -1$ *when* $yi = -1$
$H1$ and $H2$ are the planes:
$H1$: $xi \bullet w + b = +1$
H2: xi•w+b = -1

The points on the planes H1 and H2 are the Support Vectors

$d^+$= the shortest distance to the closest positive point

$d^-$= the shortest distance to the closest negative point

The margin of a separating hyperplane is $d^- + d^+$.

In order to maximize the margin, we need to minimize ||w||. With the

condition that there are no data points between H1 and H2:

$xi \bullet w + b \geq +1$ *when* $yi = +1$

$xi \bullet w + b \leq -1$ *when* $yi = -1$ Can be combined into $yi(xi \bullet w) \geq 1$

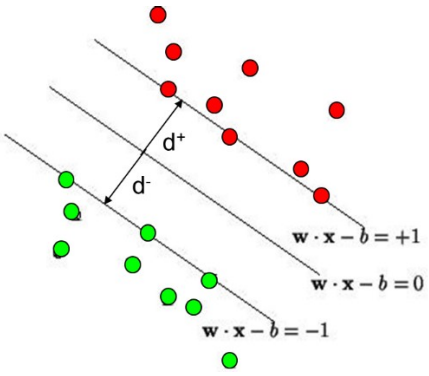



*Figure 33 The Hyperplane H and support Vectors*

choice of a hyperplane, the points x in the feature space that are mapped into the hyperplane are defined by the relation:$\sum_i \alpha_i\, k(x_i, x) = costant$. Note that if $k(x, y)$ becomes small as $y$ grows further away from $x$, each term in the sum measures the degree of closeness of the test point $x$ to the corresponding data base point $x_i$. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points $x$ mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets which are not convex at all in the original according to the equations above related to figure [33], to optimize the problem by using LaGrange method we obtain the following:

$$\text{Maximize } \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$$

$$\text{subject to } \sum_i y_i \alpha_i = 0 \text{ and } \alpha_i \geq 0$$

If the decision function is not linear then using (Kernel Trick)

Data points are linearly separable in the space $(x_1^2, x_2^2, \sqrt{2}x_1 x_2)$

$$\text{We want to maximize } \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \langle F(\mathbf{x}_i) \cdot F(\mathbf{x}_j) \rangle \text{ Define } K(\mathbf{x}_i, \mathbf{x}_j) = \langle F(\mathbf{x}_i) \cdot F(\mathbf{x}_j) \rangle$$

$$\text{the good thinge with kernel : } K \text{ is often easy to compute directly! Here, } K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle^2$$

In the SKlearn toolkit we used in our application we can set the kernel parameters and identify if the problem is linear or not.

#### *9.1.1.2. Linear Regression:*

Linear regression is a common statistical tool for modeling the relationship between some "explanatory" variables and some real valued outcome. It considered as a learning problem, the domain set $X$ is a subset of $R^d$, for some $d$, and the label set $Y$ is the set of real numbers. We would like to learn a linear function:

$H : R^d \rightarrow R$ that best approximates the relationship between our variables.

A regression problem is a prediction where the target is continuous and its applications are several, so it's important to understand how a linear model can fit the data, what its strengths and weaknesses are, and when it's preferable to pick an alternative.

The Linear regression models:

Consider a dataset of real-values vectors:

$$X = \{\overline{x_1}, \overline{x_2}, \ldots., \overline{x_n}\} \, where \, \overline{x_1} \in R^m$$

Each input vector is associated with a real value $y_i$:

$$Y = \{y_1, y_2, \ldots., y_n\} \, where \, y_n \in R$$

A linear model is based on the assumption that it's possible to approximate the output values through a regression process based on the rule:

$$\tilde{y} = \alpha_0 + \sum_{i=1}^{m} \alpha_{i\, x_i} \, where \, A = \{\alpha_0, \alpha_1, \ldots., \alpha_m\}$$

In other words, the strong assumption is that our dataset and all other unknown points lie on a hyperplane and the maximum error is proportional to both the training quality and the adaptability of the original dataset. One of the most common problems arises when the dataset is clearly non-linear and other models have to be considered (such as neural networks or kernel support vector machines as we discussed in above section).

**Example**:

Let's consider an example of binomial problem that has a small dataset built by adding some uniform noise to the points belonging to a segment bounded between -6 and 6. The original equation is:

$$y = x + 2 + n, \text{ where } n \text{ is a noise term.}$$

In the figure, there's a plot with a candidate regression function:

As we're working on a plane, the regressor we're looking for is a function of only two parameters:

$$\tilde{y} = \alpha + \beta\, x$$

In order to fit our model, we must find the best parameters and to do that we choose an

ordinary least squares approach. The loss function to minimize is:

$$L = \frac{1}{2}\sum_{i=1}^{n} ||\tilde{y} - y_i||_2^2 \; which\; becomes\; L = \frac{1}{2}\sum_{i=1}^{n}(\alpha + \beta\, x_i - y_i)^2$$

With an analytic approach, in order to find the global minimum, we must impose:

$$\begin{cases} \dfrac{\partial L}{\partial \alpha} = \displaystyle\sum_{i=1}^{n}(\alpha + \beta x_i - y_i) = 0 \\ \dfrac{\partial L}{\partial \beta} = \displaystyle\sum_{i=1}^{n}(\alpha + \beta x_i - y_i)x_i = 0 \end{cases}$$

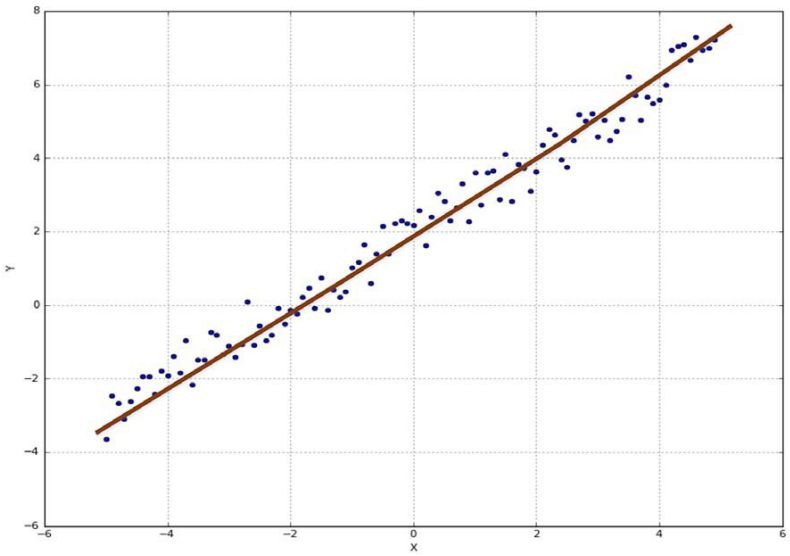



*Figure 34 bi-dimensional*

#### 9.1.1.3. Perceptron.

In machine learning, the perceptron is an algorithm for supervised learning of binary classifiers: functions that can decide whether an input (represented by a vector of numbers) belongs to one class or another. [50] It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. The algorithm allows for online learning, in that it processes elements in the training set one at a time.

In the modern sense, the perceptron is an algorithm for learning a binary classifier: a function that maps its input x (a real-valued vector) to an output value f(x) (a single binary value):

$$f(x) = \begin{cases} 1 \; if \; w \,.\, x \; > 0 \\ 0 \quad otherwise \end{cases}$$

where $w$ is a vector of real-valued weights, $w \,.\, x$ is the dot product $\sum_{i=0}^{m} w_i \,.\, x_i$ , where m is the number of inputs to the perceptron and b is the bias. The bias shifts the decision boundary away from the origin and does not depend on any input value.

The value of $f(x)$ $(0 \; or \; 1)$ is used to classify $x$ as either a positive or a negative instance, in the case of a binary classification problem. If $b$ is negative, then the weighted combination of inputs must produce a positive value greater than $|b|$ in order to push the classifier neuron over the 0 threshold. Spatially, the bias alters the position (though not the orientation) of the decision boundary. The perceptron learning algorithm does not terminate if the learning set is not linearly separable. If the vectors are not linearly separable learning will never reach a point where all vectors are classified properly. The most famous example of the perceptron's inability to solve problems with linearly non-separable vectors is the Boolean exclusive-or problem.

In the context of neural networks, a perceptron is an artificial neuron using the Heaviside step function as the activation function. The perceptron algorithm is also termed the single-layer perceptron, to distinguish it from a multilayer perceptron, which is a misnomer for a more complicated neural network. As a linear classifier, the single-layer perceptron is the simplest feedforward neural network.

### 9.1.2. Probabilistic Classifier.

#### 9.1.2.1. Naïve Byes.

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Naive Bayes are a family of powerful and easy-to-train classifiers that determine the probability of an outcome given a set of conditions using Bayes' theorem.

**Naïve Byes model:**

Abstractly, naive Bayes is a conditional probability model:

Let's assume that we have $n$ feature samples, independent variable, to be classified, represented by a vector $x$ and corresponding k classes C.

Then:

$The \; Features$: $x = (x_1, x_2, \ldots\ldots, x_n)$

$The \; Joint \; Probability$: $P(C_k|x_1, x_2, \ldots\ldots, x_n) = P(C_k)P(x_1|C_k)P(x_2|C_k, x_1) \ldots\ldots P(x_n|C_k, x_1, x_2, \ldots\ldots, x_n)$

The joint probability result from Bayesian Theorem in witch:

$$P(C_k|x_1) = \frac{P(C_k)\,P(x|C_k)}{P(x)}$$

Since the denominator is independent from $C$ then the joint probability result from chain rule are as following in equation above of joint Probability $P$ which can be simplified using conditional independence by the assumption that the features $x_i$ are independent so that:

Using conditional independence $\begin{cases} P(x_i|C_k, x_j) = P(x_i|C_k) \\ \dots \\ P(x_i|C_k, x_j, x_l, \dots, x_n) = P(x_i|C_k) \end{cases}$ then:

The joint probability can be expressed as:

$$P(C_k|x_1, x_2, \dots, x_n) \propto P(C_k) \prod_{i=1}^{n} P(x_i|C_k)$$

Then the joint probability represented as:

$$P(C_k|x_1, x_2, \dots, x_n) = \frac{1}{S} P(C_k) \prod_{i=1}^{n} P(x_i|C_k)$$

$$where\ S = P(x)\ is\ the\ scaling\ factor\ that\ depend\ only\ on\ the\ features\ x_i\ that\ considered\ as\ constant$$

## Naive Bayes classifiers

A naive Bayes classifier implement the Naïve byes theorem concluded in the previous section, although that features entry's in most cases are strictly correlated so that the assumption of the feature being dependent assumed by the Byes theorem is violated, this question was more and frequently wandering how Naïve Byes model applies to the features that are strictly correlated although it's based on the assumption of feature independence, finally I got the answer *Giuseppe Bonaccorso [51]*an Italian machine learning expert and author, saying that "in some cases ,not rare, and under particular conditions, different dependencies cancels each another, and a naive Bayes classifier succeeds in achieving very high performances even if its naiveness is violated".

If we consider a dataset:

$$X = \{\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}\}\ where\ \overline{x_i}\ \in R^m$$

Every feature vector, for simplicity, will be represented as:

$$\overline{x_i} = [x_1, x_2, \dots, x_n]$$

The output class Y:

$$Y = \{\, y_1, y_2, \dots, y_n\}\ where\ y_n \in (0,1,2, \dots, P)$$

Here, each y can belong to one of P different classes. Considering Bayes' theorem under

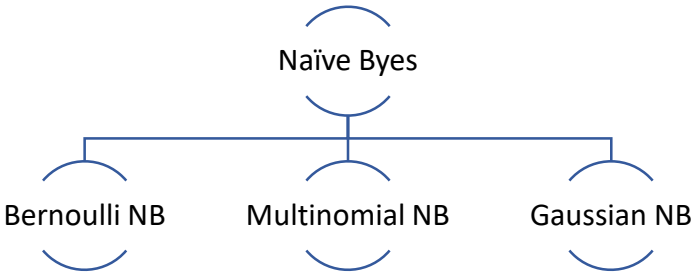
conditional independence, we can write:

$$P(y|x_1, x_2, \ldots, x_m) \propto \alpha\, P(y) \prod_{i=1}^{m} P(x_i|y)$$

The values of the probability $P(y)$ and of the conditional probabilities $P(x_i|y)$ are obtained through a frequency count, therefore, given an input vector $x$, the predicted class is the one for which the posterior probability is maximum.

**Naïve Byes implementation.**

Naïve Byes can be implemented based on the probability distribution as:




Bernoulli NB has a binary distribution feature may (present or not-present).

Multinomial NB has a discrete distribution and used whenever a feature must be represented by a whole number, in NLP, it can be the Term Frequency.

Gaussian NB has a continuous distribution characterized by its mean and variance.

We will discuss two of them that have been used in our work Multinomial NB and Gaussian NB.

### *1. Multinomial NB*

A multinomial distribution is useful to model feature vectors where each value represents, the number of occurrences of a term or its relative frequency as TFIDF or count features generated by *tfidfVictorizer* or *DictVictorizer* Features Extraction. In our work, this frequency represents the frequency of the hate terms occurrence in tweet text.

Let's assume that we have a multinomial event model, the feature vectors represent the frequencies of events occurrences, then a multinomial:

$$(P_1, P_2, \ldots, P_n)\ where\ P_i\ is\ the\ probability\ that\ event\ i\ occurs.$$

A feature vector:

$$X = (x_1, x_2, \ldots, x_n) where\ \ x_i\ the\ frequency\ of\ event\ i\ occurrences.$$

This module shows the best result in our work of hate speech detection because this event model typically used for text classification, with events representing the occurrence of a word in a single text.

Probability of observing a feature $x$:

$$P(x|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i\,!} \prod_i {P_{ki}}^{x_i}$$

### 2. Gaussian naive Bayes

This model for working on continuous values whose probabilities distribution is a Gaussian distribution:

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The conditional probabilities $P(x_i|y)$ are also Gaussian distributed; therefore, it's necessary to estimate the mean and variance of each of them using the maximum likelihood approach.

This quite easy; in fact, considering the property of a Gaussian, we get:

$$L(\mu; \sigma^2; x_i|y) = log \prod_k P(x_i^{(k)}|y) = \sum_k log\ P(x_i^{(k)}|y)$$

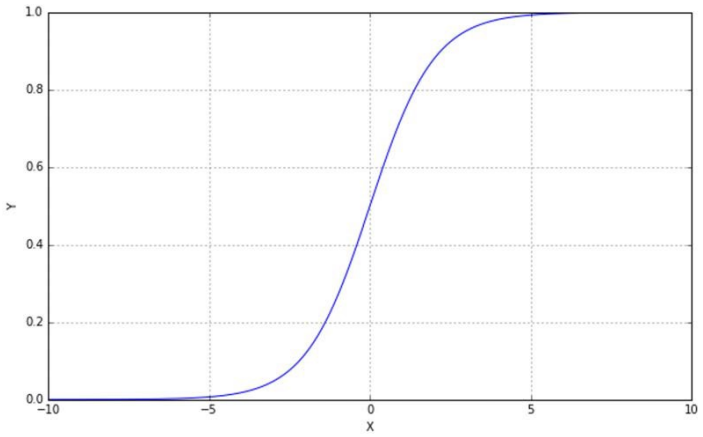
Here, the $k$ index refers to the samples in our dataset and $P(x_i|y)$ is a Gaussian itself.

## 9.1.2.2. Logistic Regression.

Even if called regression, this is a classification method which is based on the probability for a sample to belong to a class. As our probabilities must be continuous in *R* and bounded between (0, 1), it's necessary to introduce a threshold function to filter the term *z*. The name logistic comes from the decision to use the sigmoid (or logistic) function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \ which\ becomes\ \sigma(\overline{x}, \overline{w}) = \frac{1}{1+e^{-\overline{x}.\overline{w}}}$$

A partial plot of this function is shown in the following figure:




As you can see, the function intersects *x=0* in the ordinate 0.5, and *y<0.5* for *x<0* and *y>0.5* for *x>0*. Moreover, its domain is *R* and it has two asymptotes at 0 and 1. So, we can define the probability for a sample to belong to a class (from now on, we'll call them 0 and 1) as:

$$P(y|\overline{x}) = \sigma(\overline{x}, \overline{w})$$

At this point, finding the optimal parameters are equivalent to maximizing the log-likelihood given the output class:

$$L(\overline{w}, y) = \log P(y|\overline{w}) = \sum_i \log P(y_i|\overline{x_i}, \overline{w})$$

Therefore, the optimization problem can be expressed, using the indicator notation, as the minimization of the loss function:

$$J(\overline{w}) = -\sum_i \log P(y_i|\overline{x_i}, \overline{w}) = -\sum_i (y_i \log \sigma(z_i) + (1 - y_i) \log(1 - \sigma(z_i)))$$

If *y=0*, the first term becomes null and the second one becomes $log(1 - x)$, which is the log probability of the class 0. On the other hand, if $y = 1$, the second term is 0 and the first one represents the log-probability of $x$. In this way, both cases are embedded in a single expression. In terms of information theory, it means minimizing the cross-entropy between a target distribution and an approximated one:

$$H(X) = -\sum_{x \in X} P(x) log_2\, q(x)$$

In particular, if $log_2$ is adopted, the functional expresses the number of extra bits requested to encode the original distribution with the predicted one. It's obvious that when $J(w) = 0$, the two distributions are equal. Therefore, minimizing the cross-entropy is an elegant way to optimize the prediction error when the target distributions are categorical.

### 9.1.3. Decision Tree.

Decision tree is a predictive model which maps observations about an item to conclusions about the item's target value. It is a one of the predictive modelling approaches used in statistics, data mining and machine learning. It can be classified as classification tree if the target takes a finite value, and as regression tree if the target takes continuous values, in the hierarchy of the tree consists of root node. This node contains a condition that checks one of the

input value's features, and selects a branch based on that feature's value, a decision node that checks the feature values and the branches is a set of combination of features that lead to class labels and the leaves represent class label.

Practically a decision tree is a predictor, $H: X \rightarrow Y$ , that predicts the label associated with an instance $x$ by traveling from a root node of a tree to a leaf.

Decision tree is not the most common methods for classification, and it used for the problems of low complexity.

As an alternative, the *ensemble methods* are a powerful alternative to complex algorithms because they try to exploit the statistical concept of majority vote. In particular, we're going to discuss random forests of decision trees that can optimize the learning process by focusing on misclassified samples or by continuously minimizing a target loss function.

#### *9.1.3.1. Binary Decision Tree.*

In Sklearn machine learning tool we used in our application DecissionTreeClassifier implemented by binary decision tree with Gini and cross-entropy impurity index measures.

Let's consider an input dataset $X$ :

$$X = \{\overline{x_1}, \overline{x_2}, \dots\dots, \overline{x_n}\}$$

Every vector $\overline{x_i}$ is made up of n features, so each of them can be a good candidate to create a node based on the (feature, threshold) tuple, the decision is based on the condition at the node as a comparison between the feature and the threshold, the selection of the feature based on its *purity* which determine if the feature can be subdivided in the following branches so *impurity* concept is used so that an ideal scenario is based on nodes where the impurity is null so that all subsequent decisions will be taken only on the remaining features.

The goal is to reduce the residual impurity in the least number of splits so as to have a very short decision path between the sample data and the classification result, so defining the selection tubal $< i, t_i >$ with i-th feature and $t_i$ $threshould$ and apply it to the impurity measure has an impact on algorithm performance.

#### *9.1.3.2. Random Forests*

Random forests are a notion of the general technique of random decision forests [52] that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. [53]

Training can take two different approaches, a strong learner or a weak learner.

In strong learner, trained models on single instances, iterating an algorithm in order to minimize a target loss function. While in the weak learner, set of weak learners trained in parallel or sequentially and used as an ensemble based on a majority vote or the averaging of results.

These methods can be classified into two main categories:

**Bagged (or Bootstrap) trees**: In this case, the ensemble is built completely. The training process is based on a random selection of the splits and the predictions are based on a majority vote. Random forests are an example of bagged tree ensembles.

**Boosted trees:** The ensemble is built sequentially, focusing on the samples that have been previously misclassified.

**Random forests Classifier**

As we mentioned before, that class of decision trees has an infinite VC dimension. So in Random Forests, We restricted the size of the decision tree and by constructing an ensemble of trees.

“A random forest is a classier consisting of a collection of decision trees, where each tree is constructed by applying an algorithm A on the training set S and an additional random vector,$\theta$, where $\theta$ is sampled $i.i.d.$ from some distribution. The prediction of the random forest is obtained by a majority vote over the predictions of the individual trees. To specify a particular random forest, we need to define the algorithm A and the distribution over $\theta$. There are many ways to do this and here we describe one

particular option. We generate $\theta$ as follows. First, we take a random subsample from S with replacements, namely, we sample a new training set $S'$ of size $m'$ using the uniform distribution over S. Second, we construct a sequence $I_1, I_2$, ... each $I_t$ $is\ a\ subsit\ of$ $[d]$ of size $k$ where each $I_t$ is a subset of $[d]$ of size $k$, which is generated by sampling uniformly at random elements from $[d]$. All these random variables form the vector $\theta$. Then, the algorithm A grows a decision tree (e.g., using the ID3 algorithm) based on the

sample $S'$, where at each splitting stage of the algorithm, the algorithm is restricted to choosing a feature that maximizes Gain from the set $I_t$. Intuitively, if k is small, this restriction may prevent overfitting". This definition is taken by *Breiman (2001)*. [54]

A random forest can be defined also as a set of decision trees built on random section of the samples with different policy for splitting a node, rather than looking for the best choice as we seen in the decision tree selection strategy, on the previous section, in such a model, a random subset of features (for each tree) is used, trying to find the threshold that best separates the data.

As a result, there will be many trees trained in a weaker way and each of them will produce a different prediction.

The result of voting can be interpreted in two different methods, one based on the majority vote (the most voted class will be considered correct). However, in scikit-learn that we used in our work, it implements an algorithm of voting based on averaging the results of each individual voting, which yields very accurate predictions.

## 9.2. Unsupervised Learning.

Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning. [39]

## 9.3. Reinforcement learning.

Reinforcement learning is an area of machine learning inspired by behaviorist psychology, concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.

# CHAPTER 10.

# (HATE SPEECH).

## 10.1. Definition.

Hate speech, can be defined as the speech that attacks a person or group on the basis of attributes such as gender, ethnic origin, religion, race, disability, or sexual orientation. [41] [42]

Hate speech, in law, defined as any speech, gesture or conduct, writing, or display which is forbidden because it may incite violence or prejudicial action against or by a protected individual or group, or because it disparages or intimidates a protected individual or group. The law may identify a protected group by certain characteristics. [43] [44] [45] [46] In some countries, a victim of hate speech may seek redress under civil law, criminal law, or both.

The International Covenant on Civil and Political Rights (ICCPR) states that "any advocacy of national, racial or religious hatred that constitutes incitement to discrimination, hostility or violence shall be prohibited by law". [47]

## 10.2. Hate Speech Detection Difficulties.

Those definition of hate speech doesn't include the definition of offensive language because some people often use terms that are highly offensive to certain groups but in a qualitatively different manner. Those custom terms differ from country to another country from place to another place it is a tone based and culture specific generated language terms, it is evolved over time and generate a custom offensive word, as an example some African Americans often use the term nigga in everyday language online (Warner and Hirschberg 2012), people use terms like hoe and bitch when quoting rap lyrics, and teenagers use homophobic slurs like fig as they play video games. Such language is prevalent on social media (Wang et al. 2014), making this boundary condition crucial for any usable hate speech detection system.

So, the meaning can differ according to contexts of speech and habits of communities and this is a critical problem to researchers to distinguish hate speech from offensive language.

## 10.3. Hate Speech Dataset.

Dataset is the fundamental part of Supervised Machine Learning systems, the performance of the models depends on the labeled dataset correctness and accuracy of annotation, the datasets is a set of entries, collections that randomly collected and annotated so there is no guaranteed about its generalization of the domain of study ,in other words datasets samples usually doesn't encompass all of significant information related to the problem under study.as well the process of annotation and its correctness and accuracy have a direct influence in system accuracy.

Thanks to Thomas Davidson, Dana Warmsley, Michael Macy, Ingmar Weber, for offered dataset annotated by crowed flower experts for open research [40].

The dataset has been labeled by three to six experts, the process of dataset preparation has started by collection of the hate speech lexicon that contain for common hate terms compiled by hatebase.org, then by using the Twitter API they looked for tweets containing terms from this lexicon, resulting in a sample of tweets from 33,458 Twitter users. They have extracted the time-line for each user, resulting in a set of 85.4 million tweets. From this corpus, they then took a random sample of 25k tweets containing terms from the lexicon and had them manually coded by Crowd Flower (CF) workers. Workers were asked to label each tweet as one of three categories: hate speech, offensive but not hate speech, or neither offensive nor hate speech.

Crowd Flower (CF) workers of annotation have been asked to take into account the context in which tweeter users used. They were instructed that the presence of a particular word, however offensive, did not necessarily indicate a tweet is hate speech. [40]

Resulting in 24,802 labeled tweet a summarized in table [21] below:

| CLASS | NUMBER OF TWEETS | PERCENTAGE |
|---|---|---|
| HATE SPEECH | 1430 | 5.7% |
| OFFENSIVE LANGUAGE | 19190 | 77.3% |
| NEITHER | 4163 | 16.7% |
| UNANIMOUS | 19 | 0.3% |

**Table 21 Annotation Summary of Dataset of 24802 sample by CF Experts**

# CHAPTER 11.

# (FEATURES EXTRACTION).

In this section, we describe how and which features we extracted from tweet texts, and witch one we used for classification, we describe the possible features used in NLP and then we motivate our feature choice.

The main objective of our work was to find a methodology for Natural Language Feature Extraction that relays on the meaning of language and trying in somehow to transfer this meaning into a vectorization values to be fit to machine learning Classification models. So, our choice of features was motivated on this regard.

In the following sections we describe a set of feature extraction models used and the proposed (LSFET "Linguistic Structure Features Extraction Technique") as a new feature extraction relayed on the textual meaning.

## 11.1. N-Gram Features.

Now we will look at some feature-extraction concepts and techniques specially aligned towards text data.

The *Vector Space Model* is a concept known as the *Term Vector Model*, is defined as a mathematical and algebraic model for transforming and representing text documents as numeric vectors of specific terms that form the vector dimensions. In this model, it defines (Document Vector Space) in which document represented as a vector of witch each column represented by total number of distinct terms or words for all documents in the vector space:

$$VS = \{W_1, W_2, \dots, W_n\}$$

$$where\ n = distinct\ words\ across\ all\ documents$$

Where there are $n$ distinct words across all documents. Now we can represent document $D$ in this vector space as:

$$D = \{W_{D1}, W_{D2}, \dots, W_{Dm}\}$$

$$where\ D : is\ the\ document\ Number\ and\ m\ is\ the\ word\ number\ with\ in\ document\ D\ and$$

$$W_{Dm}\ is\ the\ weight\ of\ word\ m\ in\ document\ D$$

This weight can be the frequency of that word in the document, average frequency of occurrence or TF-IDF weight.

We will be talking about and implementing the following feature-extraction techniques:

- Bag of Words model
- TF-IDF model

## 11.2. Bag of Words Model

Bag-of-Words model is a simplifying model in witch text (such as a sentence, document or tweet) is represented as the bag (multiset) of its words, bag-of-words model where the frequency or occurrence of each word is used as a feature for training a classifier.

As we discussed before in VS and the weight $W_{dn}$ is equal to its frequency of occurrence of single word or (n-grams) in that document.

Bag-of-Word model implemented in scikit-learn package by CountVectorizer class.

## 11.3. TF-IDF Model

in Bag of Words model vectors are based on absolute frequencies of word occurrences. This result in a problem that words that has a frequent occurrence penalize other words that may be more interesting and effective for classification of other documents. TF-IDF tackles this problem, it stands for Term Frequency-Inverse Document Frequency, two metrics used: term frequency (TF) and inverse document frequency (IDF).

Mathematically it can be computed as the following:

$$TFIDF = TF \times IDF$$

$$IDF(t) = 1 + \log\frac{C}{1 + df(t)}$$

$$TFIDF = \frac{TFIDF}{||TFIDF||}$$

Where TF is the frequency of the word in the document as we have computed in Bag-Of-Word model,

$IDF(t)$ Represents the inverse document frequency of the Document for the term $t$, $C$ represents the count of the total number of documents in our corpus, and $df(t)$ represents the frequency of the number of documents in which the term $t$ is present.

## 11.4. Linguistic Structure Feature Extraction Technique (LSFET).

We describe extensively on first Part of structure based system analysis linguistic technique for feature extraction and how the textual meaning has been converted into a meaningful value for training and feature preparation at high level and how it fits on the low level of vectorization and the improvements on vector space dimensionality.

The novel idea of this technique is to extract the meaning of text by elaboration of the significant and affected words on the sentence that contribute more on construction of the overall meaning of the sentence and making attention on those words on feature vector representation by increasing its weights and magnitude to transfer a high level human understanding to low level in machine representation as a numerical vector as we discussed in section [10.2] Bag-of-words and section [10.3] TFI-DF. This new generated Linguistic textual feature used as new feature to our developed framework of hate speech and offensive language classification system.

## 11.5. Tweet Sentiment Based Features.

Hate, psychologically is a sentiment as feeling, hate is a negative sentiment to be precise. So, from this high abstraction of view we divide the spoken language domain into two parts: positive and negative Therefore, we believe that relying on sentiment polarity of the tweet is an important indicator of hateful tweet existence.

Hate expression is a mutual exclusion of one or more negative hate polarity terms. Although the task of detection of hate speech differs drastically from that of sentiment analysis and polarity detection, it still makes sense to use sentiment-based features as the most basic features that allow the detection of hate speech. This is because hate speech is most likely to be present in a ``negative" tweet, rather than a ``positive" one.

We have used Vader sentiment, a tool that attributes sentiment scores to sentences as well as the words of which it is composed for sentiment intensity extraction result on the score of [positive, negative, natural and compound].

## 11.6. Tweet Hate Score Ranking.

Another feature set are extracted by ranking the tweets based on its contents relevance to the hate speech by making a comparison of tweets tokens with a hate corpus set that contains for a common hate set.

The following formula used to get the tweet ranking score of relevance:

$$score = Average\ probability * Total\ length$$

$$Score = \left(\frac{Total\ Probability}{NgramsCount}\right) * Total\ Length$$

$$Score = \sum_{ngram \in tweet} \frac{Prob_{ngram}}{Ngram\ Count} * \sum_{ngram \in tweet} len(ngram)$$

Where $Prob_{ngram}$ is the probability of ngram that found in the tweet.

$Ngram\ Count$ is the total of ngram in tweet.

The summation $\sum_{ngram \in tweet} len(ngram)$ is the total length of the ngram found in the tweet.

## 11.7. Semantic Features.

Semantic features are the features that reflect in somehow the behavior of users and describe how they uses punctuation, capitalized words, and interjections, etc. hate speech on social networks and microblogging websites do not have a specific and a common use of those formation features but it may reflect in an ambiguous way the meaning of the text, as an example:

“We will be HAPPY if you do your Internship in Brazil, we made some recommendations! to your Professors in BACKGROUND! HHHH”, The tweet is obviously offensive and shows some hate, however, even if there is no explicit use of hate words, all words have positive sentiment. However, some user behavior of punctuation uses as “!” associated with ambiguous word BACKGROUND and

recommendations can give another meaning also capitalization of some words as BACKGROUND and laughing term "HHHH" can give us an opposite orientation of the whole sentence meaning.

The use of Punctuation Marks and Capitalized words can reflect in some How the hidden meaning, Punctuations organizes the sentences, clause and phrases intention and general concepts while the Capitalized words make attention and add the value to such words that considered to be the target of actual speech.so add such aforementioned features can contribute in hate speech identification.

As we have seen that in machines learning, vectorizers used to prepare the features as a numerical vector to be fit to the classifiers. It will be beneficial if features have been represented in numerical values, semantics feature make use of features as:

- N. emotions expressions.
- N. words in the tweet.
- N. quotes.
- N. exclamation marks.
- N. question marks.
- N. full stop marks.
- N. interjections.

Punctuations as ',' comma, '.' full stop, conjunctions and other phrasal punctuations and clause links words can organize the structures of the language to describe the coherent relations between sentences, clauses and phrases that help us with detection of the sources and targets of the speech as we will see later how we use these features advantage in our developed technique of Linguistic Structure Extraction in the following sections.

Although this semantic aforementioned feature can contribute on the hate speech identifications "individually", we used some of it with integration of structure based analysis system to get the extent of its use.

## 11.8. Word to Vector Model (Word2Vec).

The word2vec model [4] is one of great attract model in machine learning.in this model words represented as a vector learned by word2vec in witch this vector carries semantic meanings and are useful in a wide range of use specifically in natural language processing.

In our developed hybrid model application, we aimed to focus on the Natural Language meaning and finding in somehow to extract this meaning in feature form represented by a numerical vector values and we find that the most amazing property of these word embedding's is that somehow these vector encodings effectively capture the semantic meanings of the words. In this model, words that we know to be synonyms tend to have similar vectors in terms of cosine similarity and antonyms tend to have dissimilar vectors. Even more surprisingly, word vectors tend to reflect the real-world laws of analogy. In analogy $[Woman\ to\ queen\ as\ man\ to\ king]$ . It turns out that $[\ V(queen) - V(woman) + V(man) \approx V(king)]$ where $V()$ denotes vector of. These observations strongly suggest that word vectors encode valuable semantic information about the words that they represent.

Word2Vec has two main models: Continuous Bag of Words (CBOW) and Skip-Gram. Those models based on the word to contexts meaning relation. In the CBOW model, we predict a word given a context (a context can be a sentence, tweet, post etc.). Skip-Gram is the opposite: predict the context given a word. We will discuss briefly the two models in the following sections.

### 11.8.1. CBOW (Continuous Bag of Word Model).

continuous bag-of-word model (CBOW) is used for single to multiple context word with high network representation, on this section we will describe" single context word" as shown on figure [37]. in this model the context refers to single word therefore the model will predict only one target word given one context word same as bi-gram language model. With reference to Figure [37]:

Input layer: represents the vocabulary with size $V$. The input vector $X = \{x_1, x_2, \ldots, x_v\}$ is one-hot encoded, that is, with some $x_k = 1$ and all other $x_{\acute{k}} = 0$ for $k \neq \acute{k}$. And a Hidden Layer: with size $N$ and its nodes has a linear Activation Function $F(x) = x$. Between Input Layer and Hidden Layer, the weights are represented by the matrix $W_{v\times n} = \begin{pmatrix} w_{11} & w_{1n} \\ w_{v1} & w_{vn} \end{pmatrix}$ .for witch each word represented by an N-dimensional vector $v_w$ of the word $w$ in the input layer.

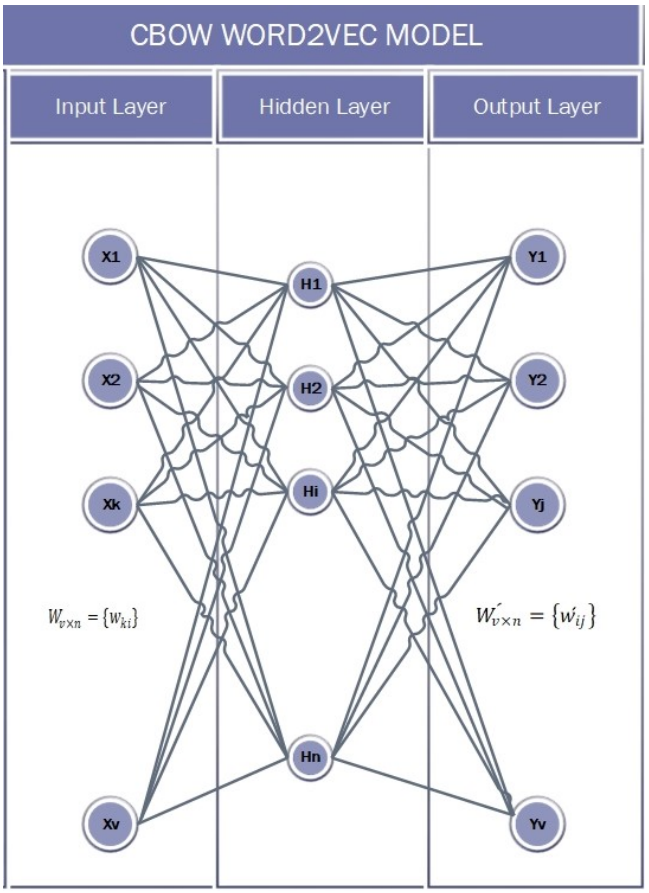



*Figure 35 CBOW MODEL*

$$h = x^T W = v_{wi} \; is \; vector \; of \; word \; wi \tag{1}$$

between hidden layer and output layer there is another weight matrix $\acute{W}_{n\times v} = \{w'_{i,J}\}$ which is an $N \times V$ matrix. Using this we can compute a score for each word in the vocabulary:

$$u_j = {v'_{wJ}}^T \cdot h \tag{2}$$

where $v'_{w_J}$ is the $jth$ column of the matrix $\acute{W}_{n\times v}$. Note that the score $u_J$ is a measure of the match between the context and the next word 1 and is computed by taking the dot product between the predicted representation ($v'_{w_J}$) and the representation of the candidate target word ($h = v_{wi}$).

By using a softmax log linear we can obtain the posterior distribution of words:

$$P(w_j|w_i) = y_j = \frac{\exp(u_j)}{\sum_{j=1}^{V}\exp(u_j)} \tag{3}$$

where $y_j$ is the output of the j-th unit of the output layer. By Substituting Equations 1 and 2 into 3 we obtain:

$$P(w_j|w_i) = y_j = \frac{\exp({v'_{w0}}^T \cdot v_{wi})}{\sum_{j=1}^{V}\exp({v'_{wj}}^T \cdot v_{wI})} \tag{4}$$

The model can be generalized to General CBOW Model as in figure [38]:

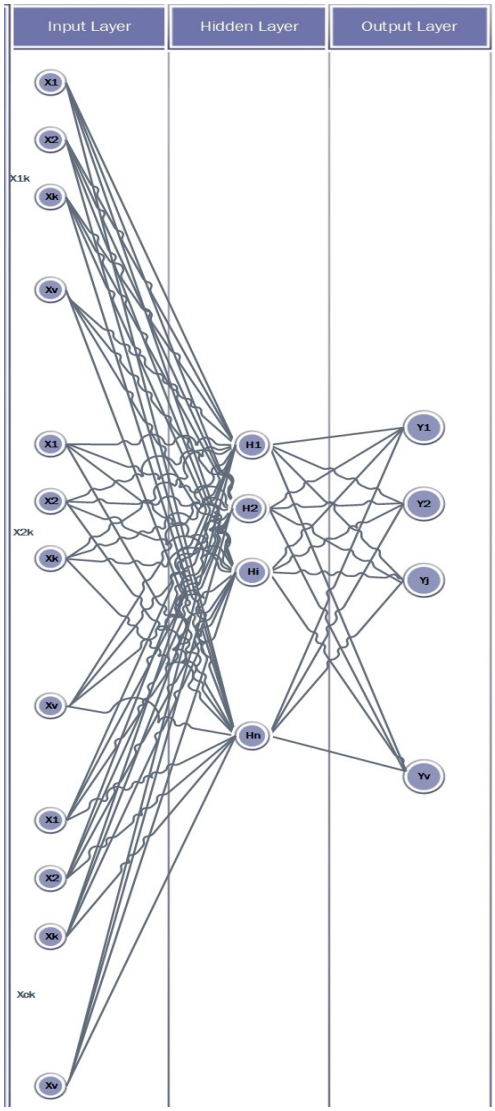



*Figure 2 General CBOW Model*

### 11.8.2. Skip-Gram Model

in Skip-Gram we predict the context C given and input word as an opposite of CBOW. Skip-Gram model objective is to predict nearby words in the associated contexts.

Skip-Gram model we input an input vector of the only word on the input layer, and as a result have the and at the output layer, instead of outputting one multinomial distribution, we output C multinomial distributions. The computations are on reverse to that of CBOW Model:

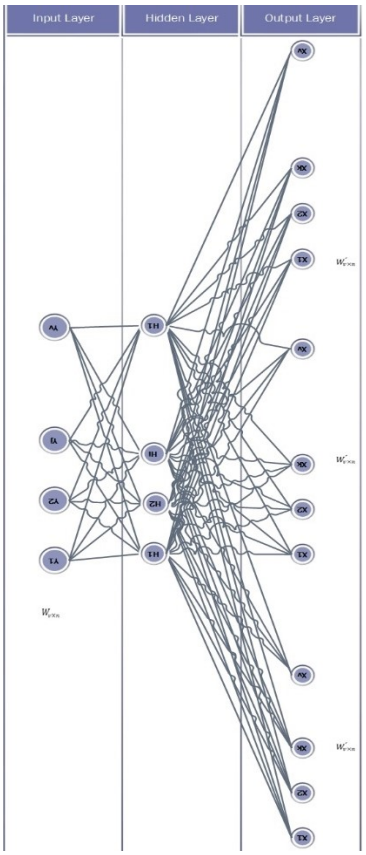



*Figure 36 Skip-Gram Model*

The final posterior probability is as following:

$$P(w_{c,j} = w_{o,c}|w_i) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j=1}^{V} \exp(u_j)} \quad (5)$$

where:

$w_{c,j}$ is the j-th word on the $c$-th panel of the output layer.

$w_{o,c}$ is the actual $c$-th word in the output context.

$w_i$ is the only input word.

$y_{c,j}$ is the output of the $j$-th unit on the $c$-th panel of the output layer.

$u_{c,j}$ is the input of the $j$-th unit on the $c$-th panel of the output layer.

Said in words, this is the probability that our prediction of the $j$-th word on the $c$-th panel, $w_{c,j}$ , equals the actual $c$-th output word, $w_{o,c}$, conditioned on the input word $w_i$.

# CHAPTER 12.

# (DIMENSIONALITY REDUCTION).

Dimensionality reduction is the first step in machine learning pipeline and the preprocessing step into classifiers, it used for removing irrelevant and redundant data, increasing learning accuracy, and improving result. In this section, some widely used feature selection and feature extraction techniques have analyzed with the purpose of visualizing how effectively these techniques can be used to achieve high performance of learning algorithms that ultimately improves predictive accuracy of classifier.

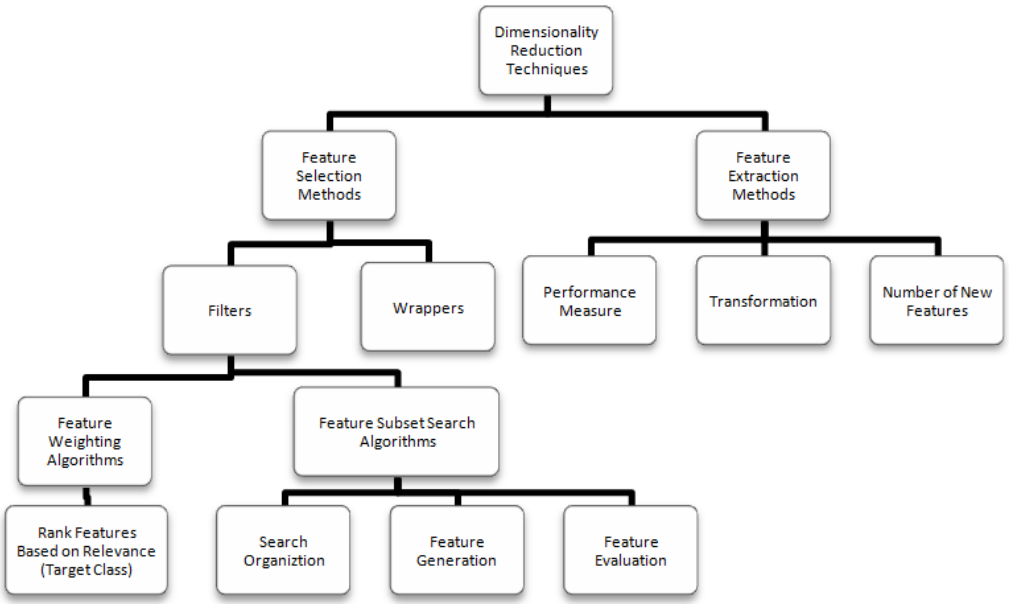



*Figure 37 Dimensionality Reduction Techniques*

## 12.1. Feature Extraction Methods in NLP

In this section, we are going to introduce two concepts in in machine learning that considered to be the first step in the learning pipeline.

Text feature extraction that extracts text information is an extraction to represent a text message

Feature extraction is the process of selecting a set of features that effectively reduce the dimension of feature space.by deleting an uncorrelated or superfluous feature. feature extraction can better improve the accuracy of learning algorithm and shorten the time. Common methods of text feature extraction include filtration, fusion, mapping, and clustering method. Traditional methods of feature extraction require handcrafted features.

**Text feature extraction methods.**

Text feature extraction plays a significant role in text classification performance and directly influencing the accuracy of text classification.

Text Feature Extraction methods is based on the VSM (vector space model),in which an n-dimensional vector space used as a space to represent texts as a point in N-dimensional Space. feature of the text represented by the Datum of each dimension of the point in the space (coordinates of the pint in the vector space) as a d (digitized) feature of the text. And the text features usually use a keyword set. It means that on the basis of a group of predefined keywords, we compute weights of the words in the text by certain methods and then form a digital vector [55]. Existing text feature extraction methods include filtration, fusion, mapping, and clustering method, which are briefly outlined below.

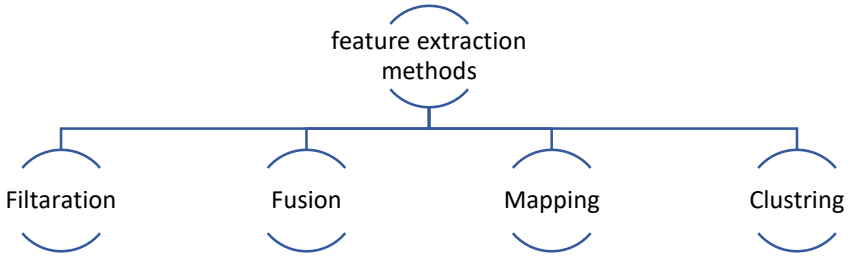



*Figure 38 Feature Extraction Methods*

### 12.1.1. Filtration

Filtration is quickly and particularly suitable for large-scale text feature extraction.

Filtration of text feature extraction mainly has word frequency, information gain, and mutual information method, etc.

#### *1. Word frequency*

Word frequency refers to the number of times that a word appears in a text. Feature selection

through word frequency means to delete the words, whose frequencies are less than a certain

threshold, to reduce the dimensionality of feature space. So as words with small frequencies have little impact on filtration [56][57][58].

#### *2. Mutual information*

MI (mutual information) [59][60] used for mutuality measurement of two objects. It is employed to measure differentiation of features to topics in filtration. Mutual information applied to represent relationships between information and the statistical measurement of correlation of two random variables [59][60]. this theory used in feature extraction is based on a hypothesis that words have big frequencies in one class than other one in a certain class but small in others. Usually, mutual information is used as the measurement between a feature word and a class, and if the feature word belongs to the class, they have the largest amounts of mutual information [60][60] .

### *3. Information gain*

IG (information gain) is the measure of information relevance to the topic under study, it is developed based on a measure of the difference between two probability distributions of the true data and the predicted data.

In filtration, it is utilized to measure the relevance of feature to topic and how much it contributes in information prediction. It can be defined as the amount of information that a certain feature item is able to provide for the whole classification. it computes information gain of each feature item with in the training data and deletes items with small information gain while the rest are ranked in a descending order based on information gain.

### *4. Impact of Filtration Methods on Text Classification.*

In our work we used two different vectorizers namely, DicVectorizer and tfidfVectorizer , that are based on DF (document frequency),IG (information gain) can reduce the dimension of vector space

model by setting the threshold and it is hard to set it, as an alternative the method MI can make the words with the lowest rising frequency get more points than by other methods,

to address low efficiency and poor accuracy of keyword extraction of traditional TF-IDF (term frequency-inverse document frequency) algorithm, a text keyword extraction method based on word frequency statistics is put forward.

## 12.1.2. Fusion method

Fusion needs integration of specific classifiers, the search conducted within an exponential time interval. As a result of its high time complexity it used for feature extraction of large-scale texts [61][62].

Fusion can be implemented by weighting method so that It gives each feature a weight within (0, 1), linear classifiers integrate weighting is highly efficient. Algorithm as K nearest neighbors, center vector weighted method are a kind of Fusion method.

### *1. K nearest neighbors*

Put forward a kind of combination of KNN classifier weighted feature extraction problem. The method is for each classification of continuous cumulative values, and it has a good classification effect. KNN method as a kind of no parameters of a simple and effective method of text categorization based on the statistical pattern recognition performance.

### *2. The center vector weighted method*

A weighted center vector classification method is proposed by Shankar [63], which firstly defines

a method of characteristics to distinguish ability, the ability to distinguish between rights and get

a new center vectors.

## 12.1.3. Mapping Methods.

In text classification Mapping methods have achieved good results [64]. It is commonly used to LSI and PCA.

### *1. Latent semantic analysis*

LSA (latent semantic analysis) uses statistical computation method to analyze a mass of text sets, thereby extracts latent semantic structure between words, and employs this latent structure to represent words and texts so as to eliminate the correlation between words and reduce dimensionality by simplifying text vectors [65].

its basic concept is that it maps a high dimensional VSM text to lower dimensional space. This mapping is achieved through SVD (singular value decomposition) of item or document matrix [66][67].

Its applications are information filtering, document index, video retrieval, text classification and clustering, information extraction.

### *2. Least squares mapping method*

It implements a high-dimensional data reduction from the perspective of center vector and least squares. He believed dimensionality reduction has its predominance over SVD, because clustered center vectors reflect the structures of raw data.

## 12.1.4. Clustering method.

Clustering takes the essential comparability of text features primarily to cluster text features into consideration. Then the center of each class is utilized to replace the features of that class.

### *1. CHI (chi-square) method*

each feature word gets a CHI value to each class, CHI clustering clusters text feature words with the same contribution to classifications, making their common classification model replace the pattern that each word has the corresponding one-dimension in the conventional algorithm.

### *2. Concept Indexing*

It is an efficient method in text classification for dimensionality reduction. The amount of classification included in training sets is exactly the dimensionality of CI subspace, which usually is smaller than that of the text vector space, so dimensionality reduction of vector space is achieved. Each class center as a generalization of text contexts in one classification can be considered as “concept,” and the mapping process of text vector can be regarded as a process of indexing in this concept space.

# CHAPTER 13.

# (SYSTEM ARCHITECTURE).

## 13.1. PROPOSED APPROACH

The aim of this work is to classify tweets into one of three classes which are:

- Hate: this class includes tweets which includes hateful tweets or offensive, and present hate, racist and segregative words and expressions.
- Offensive: this class contains tweets that are pure offensive and not belongs to hate class.
- Neither: neither hate nor offensive.

Then we extract set of features from each tweet and classify tweet into one of three classes.

## *13.2.* System *Workflow Design.*

In this section we will describe Learning Based System components, and system design of the approaches we developed

### 1. Dataset:

Data is the dominant component of the work. we described dataset collected in section [3.2.4.] [Hate Speech Dataset].

### 2. Preprocessing:

#### *A. Clean tweets.*

Preprocessing started by clean up the tweets by first removal of URLs started by ‘www’ or without it and includes ‘‘https://’’, then users tag removal i.e. the users start by @ ‘‘@user’’ and removing irrelevant expressions as white spaces , hashtags, stopwords and punctuations. Note that stopwords as verb to be or to have, conjunctions and punctuations are removed in this preprocessing as an opposed to structure based system analysis that considers stopwords and punctuations as a basic dependency relation used to link and extract the clauses entities in and between sentences.

#### *B. Tokenization.*

The second step consists of the tokenization and the lemmatization of the different words.

### 3. Features extraction.

On chapter 6 we described a set of features used in our system and we motivated our selection of those that layout on semantic meanings as:

- Tweet N-Gram Features.
- Bag of Words Model.
- TF-IDF Model.
- Linguistic Structure Feature Extraction System (LSFES) Features.
- Tweet Sentiment Based Features.

- Tweet Hate Score Ranking.
- Semantic features.
- Word to Vector Model (Word2Vec).

### 4. Classification.

On chapter 4 we described a set of classifiers used in our model and we have selected a set of supervised classifiers using the sklearn toolkit and from different classes as Linear Classifiers, Probabilistic Classifier and Decision Tree for better understand the performance of features used among the classifiers we used:

- Linear Classifiers
    - Support Vector Machine.
    - Linear Regression:
    - Perceptron.
- Probabilistic Classifier.
    - Naïve Byes.
        - Multinomial NB
        - Gaussian naive Bayes
    - Logistic Regression.
- Decision Tree.
    - Binary Decision Tree.
    - Random Forest.

### 5. Performance Evaluations.

For evaluation of classification performance, we used four different key performances indicators (KPIs) which are F1-score, the precision and recall. We will describe it later on hybrid system evaluation.

## *13.3.* System *Design Algorithm.*

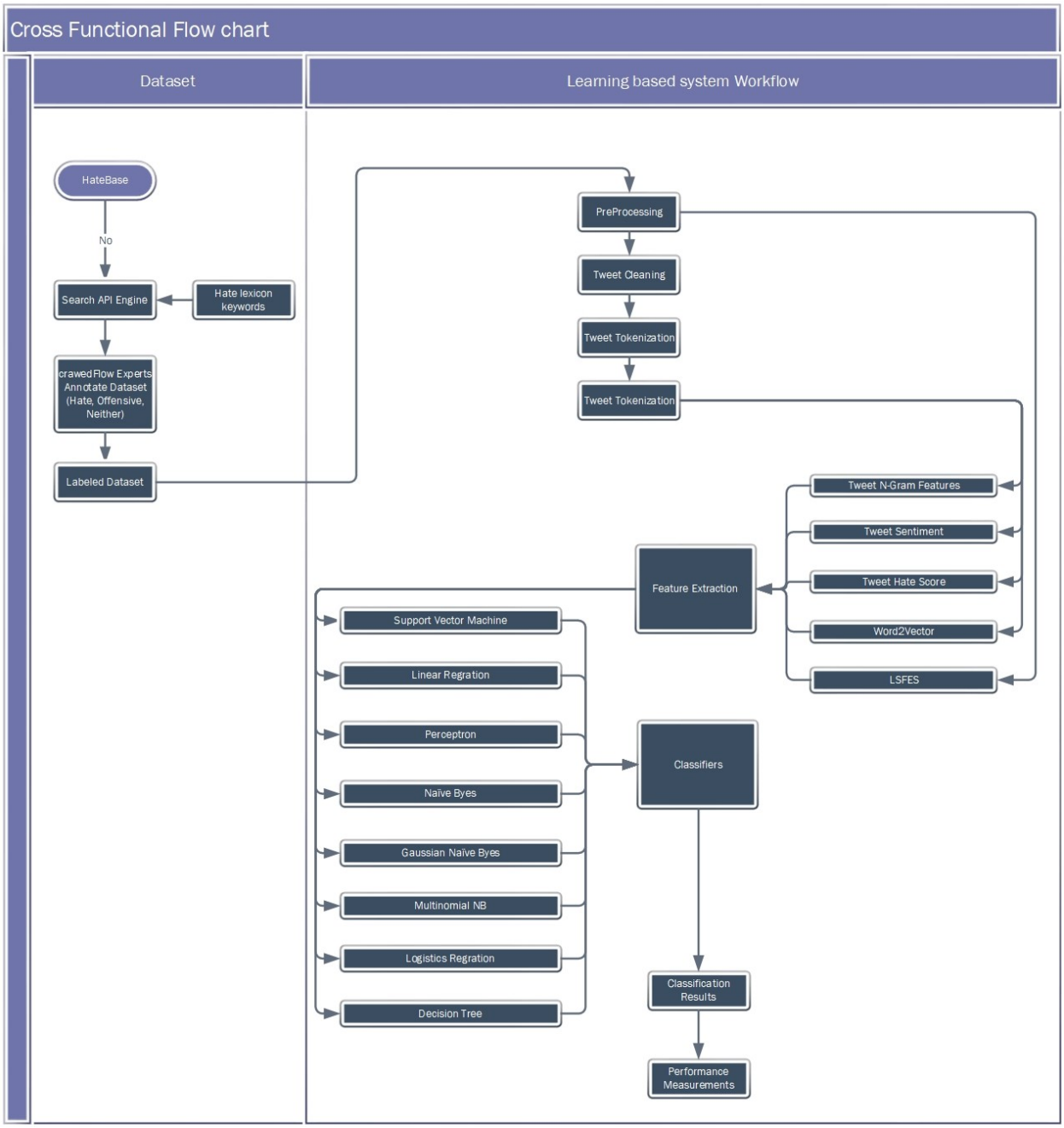



*Figure 39 Learning Based System Flowchart*

# IV. FOURTH PART (HYBIRD STRUCTURE BASED AND LEARNING BASED HATE SPEECH AND OFFENSIVE LANGUAGE CLASSIFICATION SYSTEM)

This part is an integration of the two systems described in second and third Part, in which learning based system use the feature generated by structure based systems (LSFES features) by provided interface to improve classification performance. In this part chapter [13-14] we describe and discuss a hybrid methodology and the and the compatibility between them and the final system Results.

# CHAPTER 14.

# (DEVELOPED WORKFLOW METHODOLOGY)

We Summarize the developed hybrid Structure Based and Learning Based Hate Speech and Offensive Language System components with the following characteristics:

- Tweets first preprocessed and cleaned without stop word removal.
- Structured based system takes as an input the preprocessed tweets and Extract Linguistics textual feature (Natural Language Hate Speech Templates Extraction) as pairs and triples features.
    - Natural Language Hate Speech Templates Structure Extraction by using the Consistency Grammars and Parse Tree to be used in Linguistic Analysis.
    - Generate a list of Template to be used as feature template
    - Exploiting the Dependency parse tree and POS for Analysis using Stanford CoreNLP Tools for further analysis and Linguistics Relation Extraction.
    - Development of Heuristics algorithms for each template Generated in the previous analysis stages on top of the dependency parse tree.
    - Heuristics algorithms has those functionality:
        - Dependency Parsing.
        - Traversing the dependency outputs and apply set of rules for structured templates extraction.
        - Graph generation of Dependency Parse Tree.
    - The output of this system is a NL features represented in pairs and triples as an interface to ML system.
- Learning Based System with multiple classifiers are used.
    - Preprocessing the tweets text.
        - Cleaning the tweets.
        - Stop word removal.
        - Tokenization.
        - Stemming.
        - Lemmatization.
    - Feature extraction:
        - The output of the structured based system (LSFES Features).
        - Cleaned Tweets.
        - POS.
        - Tweet Sentiment.
        - Word2vector.
        - Score of tweet contains common hate terms.
    - Classifiers used:
        - Multinomial Naive Byes classifier.
        - Support Vector Machine classifier.
        - Random Forest Classifier.
        - Perceptron classifier.
        - LinearSVC classifier.

- DecisionTree classifier.
- LinearRegression classifier.
- GaussianNB classifier.
- LogisticRegression classifier.

- Performance Measurement Evaluation.
  - F1-measure, Recall and precision.

## 14.1. Tools and utilities.

This section used to describe the tools and utilities used to build this system.

### 14.1.1. Programming Language.

Python is a widely used general-purpose scripting language, high-level programming language.[25][26] and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java.[27][28] The language provides constructs intended to enable clear programs on both a small and large scale.[29]

it supports multiple programming paradigms, including procedural programming , functional programming and object-oriented programming . It supports a dynamic type and automatic memory management.

Python has been recently associated with Data Analysis6 due to its powerful built-in idioms for

data processing and its clean syntax. [30]

Python is a widely used for Natural Language Programming due to its encompassing of large standard library tools libraries and packages .[31]

### 14.1.2. NLTK

Natural Language Toolkit (NLTK)[32] is a Python package for natural language processing,

it provides interfaces to over 50 corpora and lexical resources such as WordNet [33], along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

NLTK provides an interface text processing and linguistic structure analysis. NLTK includes libraries and programs of NLP components such as tokenization, POS tagging, parsing, chunking, semantic analysis, classification and clustering.

NLTK can be utilized for analyzing natural language and knowledge extraction or Relation Extraction, one of the application of information extraction by extracting data from helicopter maintenance records to populate a database using NLTK for partial parsing of records using hierarchical text chunking by McKenzie et al. (2010). Another developed system for question answering use NLTK for questions analysis linguistically by Stoyanchev et al. (2008). Moreover, an approach developed by Sætre (2006) for finding a biological relevant information on protein interactions from the web.

### 14.1.3. Stanford CoreNLP

Stanford CoreNLP[34], a Java or JVM-based annotation pipeline framework, which provides most of the common core natural language processing (NLP) steps starting with tokenization ,Sentence Splitting, Lemmatization, Parts of Speech, Named Entity Recognition , RegexNER, Constituency Parsing, Dependency Parsing, Coreference Resolution, Natural Logic, Open Information Extraction,Relation

Extraction ,Quote Annotator, CleanXML Annotator, True case Annotator and Entity Mentions Annotator.as in figure [42]

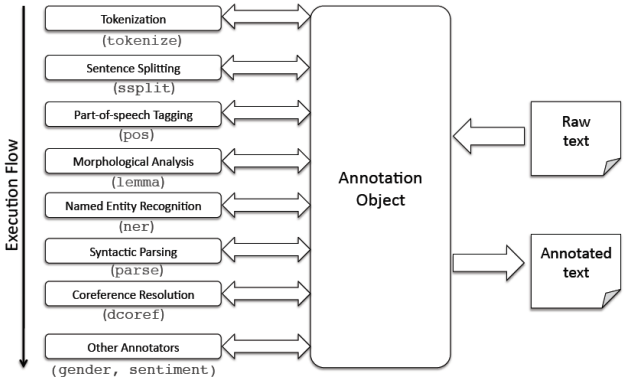



*Figure 40 Overall system architecture: Raw text Overall system architecture:*

*Raw text is put into an Annotation object and then a sequence of Annotators adds information in an analysis pipeline.*

Stanford CoreNLP [35] is an integrated framework of linguistic tools, the pipeline provides an interfaces to the individual tools and to provide a convenient framework to the developer using the utilities of the entire system and this what distinguish it from other pipeline tools , is done in this way with the intent of facilitate the creation of pipelines in which more fundamental tools are executed earlier in the process, generating output in which other of these tools build upon. In the CoreNLP each of these tools are called annotators. It provides the following annotators out of the box: Tokenization; Sentence Splitting; Lemmatization; Parts of Speech; Named Entity Recognition (described further in this document in Section 2.1); RegexNER (Named Entity Recognition); Constituency Parsing; Dependency Parsing (also in Section 2.1);Coreference Resolution; Natural Logic; Open Information Extraction (Section 3.2); Sentiment; Relation Extraction (Section 2.2); Quote Annotator; CleanXML Annotator; True case Annotator; Entity Mentions Annotator.

### 14.1.4. Scikit-Learn

Scikit-Learn is a Python module integrating classic machine learning algorithms in the tightly-knit scientific Python world (numpy, scipy, matplotlib). It aims to provide simple and efficient solutions to learning problems, accessible to everybody and reusable in various contexts: machine-learning as a versatile tool for science and engineering.[85]

We have used sklearn toolkit for classifiers implementation and performance measurement.

### 14.1.5. Gensim

Gensim is a vector space and topic modeling toolkit implemented in python. It has a rich set of capabilities for semantic analysis. Gensim includes implementations of tf–idf, random projections, word2vec and document2vec algorithms,[86] hierarchical Dirichlet processes (HDP), latent semantic analysis (LSA) and latent Dirichlet allocation (LDA), including distributed parallel versions.[88]

The important future of genism we used is the Google's very popular word2vec model, a neural network model implemented to learn distributed representations of words where similar words (semantic) occur close to each other.

### 14.1.6. Graphviz

Graphviz [89] is an open source graph generation software. Graphviz (short for Graph Visualization Software) is a package of open-source tools initiated by AT&T Labs Research for drawing graphs specified in DOT language scripts which is used to generate the graphs using the distributed DOT binary.

### 14.1.7. Java virtual machine

A Java virtual machine (JVM) is an abstract computing machine that enables a computer to run a Java program. There are three notions of the JVM: specification, implementation, and instance. The specification is a document that formally describes what is required of a JVM implementation. Having a single specification ensures all implementations are interoperable. A JVM implementation is a computer program that meets the requirements of the JVM specification. An instance of a JVM is an implementation running in a process that executes a computer program compiled into Java bytecode.

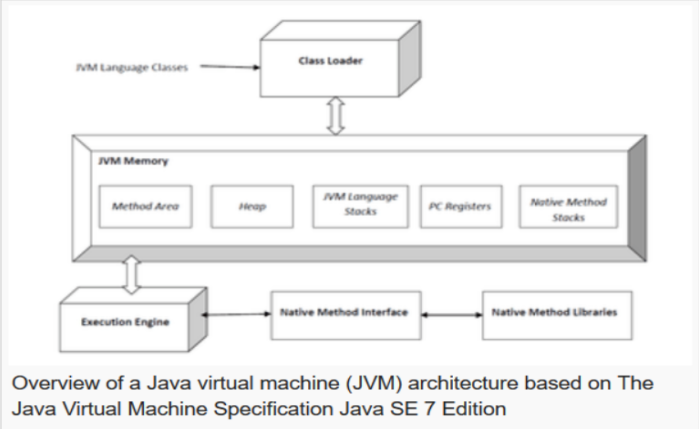



**Figure 41 JVM ARCHITECTURE**

## 14.2. Developed Workflow.

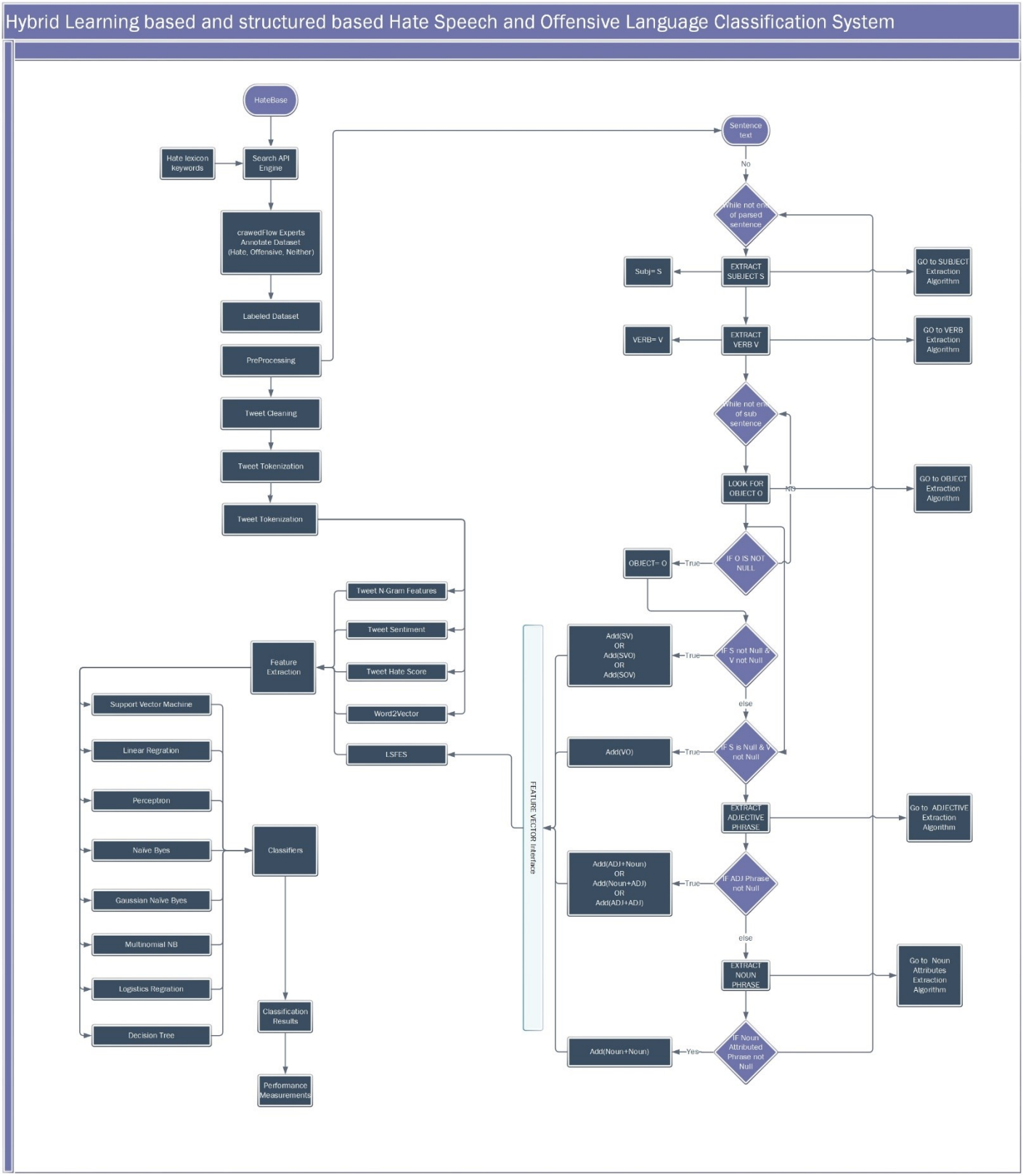



**Figure 42 Hybrid Learning Based and Structure based hate speech and offensive language classification System**

# CHAPTER 15.

# (RESULTS AND DISSCUSSION).

After we setup many experiments models with different configurations parameters setting of machine learning models and feature preparation we have got the best performing model has an overall average precision 0.97, recall of 0.97, and F1 score of 0.97 of Multinomial Naïve Byes classifier as in table [20]. The diffusion matrix shows that 30% of Hate class is misclassified and 25% moved to the offensive class and this should be happened because hate speech is an offensive language that by its context show hate intensions, also it was difficult by experts to classify hate from offensive during dataset annotation and labeling, there are 1430 hate tweet out of 25000 dataset tweet around 5.7% represent hate class as we have seen in on chapter 9 section [9.4] table [].

| Hate Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Hate Speech** | 0.93 | 0.72 | 0.81 | 137 |
| **Neither** | 0.96 | 0.97 | 0.97 | 470 |
| **offensive language** | 0.98 | 0.99 | 0.98 | 1872 |
| | | | | |
| **AVG / Total** | 0.97 | 0.97 | 0.97 | 2479 |

**Table 22 Classification Report Of Multinomial Naïve Byes with Test Score: 0.971**

We have added to each tweet a confidence number corresponding to number of coder that classify this tweet on its class to total number of coders, this confidence number represent probability of tweet to be classified or predicted on its class. annotators have annotated hate tweets if it contains a strong hate content as racial or homophobic slurs such tweets have a higher probability to be classified as hate on other side tweets that has offensive terms related to sexist targeted to girls has a higher probability to be classified as offensive. offensive class has a broad definition and it includes hate speech that can be distinguished by context.

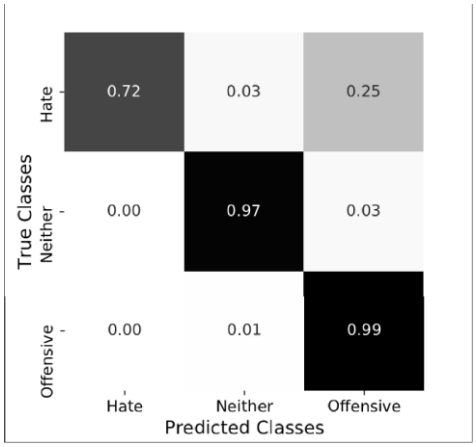



**Figure 43 Multinomial Naïve Byes confusion matrix of (Hate, Neither and offensive )**

The second good performing model obtained by Perceptron classifier a neural network classification model, it has an overall average precision 0.91, recall of 0.88, and F1 score of 0.89. The true classes with respect to predicted class presented by diffusion matrix shows that 24% of miss classified tweets moved to offensive for the same reasons explained above as shown in figure [38].

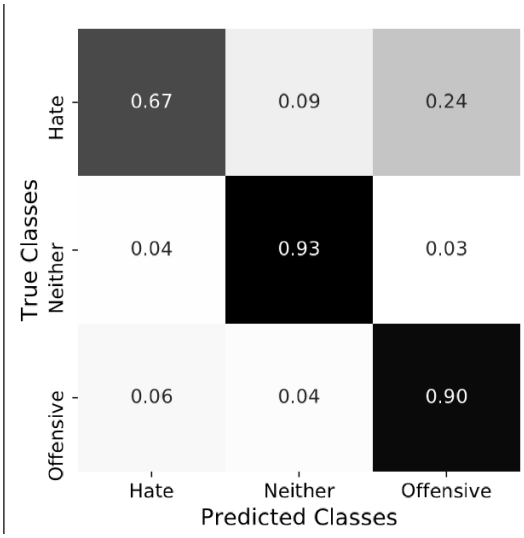



**Figure 44 Perceptron confusion matrix of (Hate, Neither and offensive )**

If we take a look to machine learning models that we used we will see that some models work better than other and this result from some compatibility between classifiers models used and features representation as we discussed in chapter 8 section [8.1.2.1] probabilistic classifier multinomial Naïve byes model we have seen that this classifier works better in Natural language processing with TFIDF or count vectorizers that represent feature vectors in a numerical values corresponding to terms frequency that can be modeled as event occurrence in the probabilistic byes model. So in our thesis we supposed that engineering behind Feature Extraction, Vectorization and Dimensionality Reduction have a direct influence on machine learning Models accuracy and efficiency, so we make some attention on those three related technique throughout system development of (LSFES) and (Learning Based System) as we described in chapter [11] and [12].

# BIBLIOGRAPHY


[1]. Hovy, Z. W., & Zeerak, W. (2016 an Diego, California, USA, June. Associati). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In Proceedings of the NAACL Student Research Workshop, 88–93.

[2]. Burnap, P., & L. Williams, M. (2016). identifying cyber hate on twitter across multiple protected characteristics. EPJ Data Science, 5(1):1–15.

[3]. Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016 ). Abusive language detection in online user content. In Proceedings of the 25th International Conference on World Wide Web. Geneva, Switzerland, 145–153.

[4]. Hee, C. V., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., Pauw, G. D., . . . Hoste, V. (2015). Detection and fine-grained classification of cyberbullying events. In Proceedings of Recent Advances in Natural Language Processing, Proceedings. Hissar, Bulgaria, 672–680.

[5]. Hosseinmardi, H., Mattson, S. A., Ibn Rafiq, R., Han, R., Lv, Q., & Mishra, S. (2015). Detection of cyberbullying incidents on the instagram social network. . CoRR, abs/1503.03909.

[6]. Xu, J.-M., Jun, K.-S., Zhu, X., & Bellmore, A. (2012 Montr´eal, Canada. Association for Computational Linguistics). Learning from bullying traces in social media. In Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies, 656–666, .

[7]. Warner, W., & Hirschberg, J. (2012. Stroudsburg, PA, USA. Association for Computational Linguistics). Detecting hate speech on the world wide web. In Proceedings of the Second Workshop on Language in Social Media, LSM ’12, 19–26.

[8]. Sood, S. O., F. Churchill, E., & Antin, J. (2012b). Automatic identification of personal insults on social news sites. J. Am. Soc. Inf. Sci. Technol, 63(2):270–285, February.

[9]. Catherine, M., de Marne_e, & D. Manning, C. (September 2008). Stanford typed dependencies manual.

[10]. Mehdad, Y., & Tetreault, J. (2016). Do characters abuse more than words. In 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 299–303, Los Angeles, CA, USA.

[11]. M. Blei, D., Ng, A., & I. Jordan, M. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3:993–1022.

[12]. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at the International Conference on Learning Representations (ICLR), Scottsdale, AZ, USA.

[13]. Gitari, N. D., Zuping, Z., Damien, H., & Long, J. (2015). A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering, 10(4):215–230.

[14]. Greevy, E., & F. Smeaton, A. (2004). Classifying racist texts using a support vector machine. In Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 468–469. ACM.

[15]. Dinakar, K., Reichart, R., & Lieberman, H. (2011). Modeling the detection of textual cyberbullying. The Social Mobile Web. 11(02).

[16]. Angeli, G., Johnson Premkumar, M., & D.Manning, C. (2015). Leveraging Linguistic Structure For Open Domain Information Extraction. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Beijing, China: Association for Computational Linguistics, 344–354.

[17]. Etzioni et al, O. (2011). Open Information Extraction: The Second Generation. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One. IJCAI'11.Barcelona, Catalonia, Spain: AAAI Press, 3–10. doi:10.5591/978-1-57735-516-8/IJCAI11-012.

[18]. Del Corro, L., & Gemulla, R. (2013). ClausIE: Clause-based Open Information Extraction. In: Proceedings of the 22Nd International Conference on World Wide Web. WWW '13. Rio de Janeiro, Brazil: ACM, 355–366.

[19]. Chen, Y. (2011). Detecting Offensive Language in Social Medias for Protection of Adolescent. PhD thesis, The Pennsylvania State University.

[20]. ack Mondal, M., AraújoSilva, L., Horizonte, B., & Benevenuto, F. (2017). A Measurement Study of Hate Speech in Social Media.

[21]. (2002:3), L., & (2004: 14, 348), R. ((1997:264)). For definitions and discussions of the noun phrase that point to the presence of a head noun, see for instance Crystal.

[22]. Clahsen, Felser, Harald, & Claudia. (n.d.). Grammatical Processing in Language Learners. Applied Psycholinguistics, 3-42. doi:10.1017/S0142716406060024

[23]. Merriam. (Incorporated. 2014). Merriam-Webster Dictionary (online).

[24]. Capital Community College Foundation. (Retrieved 20 March 2012).

[25]. TIOBE Programming Community Index Python. (Retrieved 10 September 2015). TIOBE Software Index.

[26]. ""The RedMonk Programming Language Rankings: June 2015 – tecosystems. (2015). Redmonk.com. 1 July 2015. Retrieved 10 September 2015.

[27]. Summerfield, &, M. (n.d.). Rapid GUI Programming with Python and Qt. Python is a very expressive language, which means that we can usually write far fewer lines of Python code than would be required for an equivalent application written in, say, C++ or Java.

[28]. McConnell, & Steve. (n.d.). Code Complete ,2009.

[29]. Kuhlman, & Dave. (n.d.). A Python Book: Beginning Python, Advanced Python, and Python Exercises.

[30]. https://www.quora.com/Why-is-Python-a-language-of-choice-for-data-scientists. (n.d.).

[31]. About Python". Python Software Foundation. Retrieved 24 April 2012., second section "Fans of Python use the phrase "batteries included" to describe the standard library, which covers everything from asynchronous processing to zip files. (n.d.).

[32]. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.

[33]. A. Miller, G. ((Nov. 1995)). 'WordNet: A Lexical Database for English'. In: Commun. ACM 38.11 .

[34]. D. Manning, C. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations, 55–60.

[35]. D.Manning, & Christopher. (n.d.). The Stanford CoreNLP Natural Language Processing Toolkit. Linguistics & Computer Science Stanford University ,MihaiSurdeanu SISTA ,JohnBauer Dept of Computer Science Stanford University ,JennyFinkel Prismatic Inc. ,StevenJ.Bethard Computer and Information Sciences U. of Alabama at Birmingham ,DavidMcClosky IBM R.

[36]. V. Punyakanok, & D. Roth. (2001). The use of classifiers in sequential inference. In NIPS. MIT Press, 995–1001.

[37]. Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning. The MIT Press ISBN 9780262018258.

[38]. http://www.britannica.com/EBchecked/topic/1116194/machine-learning This is a tertiary source that clearly includes information from other sources but does not name them. (n.d.).

[39]. Jordan, M., Bishop, & Christopher M. (n.d.). "Neural Networks". In Allen B. Tucker Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, FL: Chapman & Hall/CRC Press LLC. Section VII: Intelligent Systems.

[40]. Davidson, T., Warmsley, D., Michael , M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. arXiv:1703.04009v1 [cs.CL] 11 Mar 2017.

[41]. (2011). Definitions for "hate speech", Dictionary.com. Retrieved 25 June 2011.

[42]. Nockleby, John T, W. Levy , L., & L. Karst, K. (2008). "Hate Speech," in Encyclopedia of the American Constitution,ed. Leonard W. Levy and Kenneth L. Karst, vol. 3. (2nd ed.), Detroit: Macmillan Reference US, pp. 1277-1279. Cited in "Library 2.0 and the Problem of Hate Speech," by Margaret Brown-Sica and Jeff.

[43]. Criminal Justice Act 2003. (2003).

[44]. An Activist's Guide to The Yogyakarta Principles; p125 by Yogyakarta Principles in Action. (n.d.).

[45]. Kinney, & Terry A. (2010). Hate Speech and Ethnophaulisms. The International Encyclopedia of Communication. Blackwell Reference Online,Retrieved 10 March 2010. doi:10.1111/b.9781405131995.2008.x

[46]. Uslegal.com: Hate speech Retrieved 31 July 2012. (2012).

[47]. UK-USA: The British Character of America. (n.d.).

[48]. David , A. (n.d.). Statistical Models: Theory and Practice. Cambridge University Press, 128.

[49]. Walker, SH, Duncan, & DB. (1967). Estimation of the probability of an event as a function of several independent variables. Biometrika 54, 167–178.

[50]. Y , F., Schapire , & R. E. (n.d.). Large margin classification using the perceptron algorithm. Machine Learning, 37 (3): 277–296. doi:10.1023/A:1007662407062.

[51]. Giuseppe , B. (n.d.). is a machine learning and big data consultant with more than 12 years of experience He has an M.Eng. in electronics engineering from the University of Catania, Italy, and further postgraduate specialization from the University of Rome, Tor Vergata, Italy. Rome Italy.

[52]. Ho, T. K. (August 1995). Random Decision Forest. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16, 278–282.

[53]. Hastie, Trevor, Tibshirani, Robert, Friedman, & Jero. (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5.

[54]. Breiman, & Leo. (n.d.). "Random Forests". Machine Learning 45 (1): 5–32. doi:10.1023/A:1010933404324.

[55]. Mladenic, D. (n.d.). Machine learning on non-homogeneous, distributed text data, PhD Thesis. Web. 1998.

[56]. Singh V, Patnaik T, & Kumar B. (2013). Feature extraction techniques for handwritten text in various scripts: asurvey. International Journal of Soft Computing and Engineering, 3(1):238–241.

[57]. S Niharika, Latha, V., & Lavanya, D. (2006). A survey on text categorization. Int. J. Compt. Trends Technol. 3(1), 39-45 .

[58]. Mhashi , M., Mili , H., & Rada, R. (1992). Word Frequency Based Indexing and Authoring[M] Computers and Writing. Springer, Netherlands, 1992, 131-148.

[59]. L, P. (2003). Estimation of entropy and mutual information. Neural Comput. 15(6):1191–1253. doi:10.1162/089976603321780272

[60]. Russakoff , D., Rohlfing , T., & Tomasi , C. ( May 11-14, 2004). Image Similarity Using Mutual Information of Region. Computer Vision - ECCV 2004, European Conference on Computer Vision. Proceedings. (DBLP, 2004), 596-607.

[61]. Chen, S., Z , L., & H , G. (2016). An entropy fusion method for feature extraction of EEG. Neural Comput. Appl. 1-7.

[62]. K, U., & Kobayashi , T. (2007). Fusion-based age-group classification method using multiple twodimensional feature extraction algorithms. Ieice Transactions on Information and Systems, 923–934. doi:10.1093/ietisy/e90-d.6.923

[63]. Shankar, S., & Karypis, G. (2000). Weight adjustment schemes for a centroid based classifier. 1–20.

[64]. JL, S., & Blattner , F. (1978). Least-squares method for restriction mapping. Gene. PubMed] [Cross Ref], 167–174. doi:10.1016/0378-1119(78)90028-8

[65]. Evangelopoulos, N. (2013). Latent semantic analysis. Annual Review of Information Science and Technology. . 4(6):683–692.

[66]. Y , Y., & JO , P. (1997). A Comparative Study on Feature Selection in Text Categorization[C]. Fourteenth International Conference on Machine Learning. (Morgan Kaufmann Publishers Inc, 412-420.

[67]. Zhou, Y., Y , L., & S , X. (2009). an improved KNN text classification algorithm based on clustering. J. Compt. 4(3), 230-237.

[68]. I ntroducing English Linguistics International Student Edition by Charles F. Meyer. (n.d.).

[69]. Russell , T. (1986). "Basic Word Order: Functional Principles", Croom Helm, London,. 22.

[70]. Conner, S. ((1968:43ff.)). for a discussion of the traditional subject concept.

[71]. The division of the clause into a subject and a predicate is a view of sentence structure that is adopted by most English grammars. (n.d.). e.g. Conner (1968:43), Freeborn (1995:121), and Biber et al. (1999:122).

[72]. See , T. (1969). for the alternative concept of sentence structure that puts the subject and the object on more equal footing since they can both be dependents of a (finite) verb. 103-105.

[73]. de Marnee, M. C., & D. Manning, C. (2011). Stanford typed dependencies manual. September 2008 Revised for Stanford Parser v. 1.6.9 in September 2011.

[74]. Morenberg, M. (1997). Doing Grammar, Oxford University Press. 6-14.

[75]. Morenberg, M. (1997). Doing Grammar, Oxford University Press. 9-10.

[76]. Morenberg, M. (1997). Doing Grammar, Oxford University Press. 7.

[77]. Bland, K., & (1996:415. (1995). For descriptions of the traditional distinction between subject and object, see for instance Freeborn. 31.

[78]. (1968:43), C., (1995:121), F., & (1999:122), B. (n.d.). The division of the clause into a subject and a predicate is a view of sentence structure that is adopted by most grammars,.

[79]. Concerning the fact that the object is part of the predicate, see for instance Biber et al. (1999:122). (19999). 122.

[80]. Keenan , & Comrie. (1977). The insight that the arguments and adjuncts of verbs are ranked is expressed as the Accessibility Hierarchy. See Keenan and Comrie (1977).

[81]. Cobuild, C. (1995). The distinction between transitive and intransitive verbs is acknowledged by most any grammar. See for instance the Collins Cobuild Grammar (1995:139ff.).

[82]. D. Huddleston, R., & Geoffrey , K. (CUP 2005). A Student's Introduction to English Grammar. 122ff.

[83]. "Adjectives". Capital Community College Foundation. Capital Community College Foundation. Retrieved 20 March 2012. (2012).

[84]. Jackendoff, R., & Ray. (2002). "§5.5 Semantics as a generative system". Foundations of language: brain, meaning, grammar, evolution (PDF). Oxford University Press. ISBN 0-19-827012-7.

[85]. scikit-learn user guide Release 0.12-git. (2016).

[86]. Deep learning with word2vec and gensim. (n.d.). 2016.

[87]. Hovy, Z. W., & Zeerak, W. (2016 an Diego, California, USA, June. Associati). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In Proceedings of the NAACL Student Research Workshop, 88–93.

[88]. Řehůřek , R., & Petr , S. (2010). Software framework for topic modelling with large corpora. Proc. LREC Workshop on New Challenges for NLP Frameworks.

[89]. http://www.graphviz.org/. (n.d.).

[90]. Williams, P. B. (2016). identifying cyber hate on twitter across multiple protected characteristics. EPJ Data Science, 5(1), 1-15.