Lexicon based Sentiment Analyzer for the Telugu Language

Rajesh Burla, B.Tech

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Intelligent Systems)

Supervisor: Prof. Khurshid Ahmad

August 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Rajesh Burla

August 14, 2019

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Rajesh Burla

August 14, 2019

Acknowledgments

With great honor and sense of gratitude, I would like to express my deepest appreciation to my supervisor, Prof. Khurshid Ahmad, whose constant encouragement was vital in making this dissertation a reality. I have been extremely lucky to have a supervisor who provided insightful discussions about the research and who responded to my questions and queries so promptly despite his busy agenda. Thanks to his amazing guidance, perfect time management, endless encouragement, and consistently wise advice, he was a true source of inspiration. It was a real privilege and a great pleasure for me not only to learn from his exceptional knowledge but also his extraordinary human qualities.

I would like to express my sincere thankfulness to my second reader Prof. Vincent Wade who gave me suggestions for improving my thesis during my presentation.

Finally, I would like to express my deepest honor and most heartfelt thanks to my family for their warm love, continued patience, and endless support. Also, I thank my dearest friends for encouraging me to be an independent thinker and having confidence in my ability to go after new things that inspired me.

Rajesh Burla

University of Dublin, Trinity College
August 2019

Lexicon based Sentiment Analyzer for the Telugu Language

Rajesh Burla, Master of Science in Computer Science
University of Dublin, Trinity College, 2019

Supervisor: Prof. Khurshid Ahmad

Digital media includes social media, blogs and diverse data that provides instantaneous access to information, now and can be any time in history. This data in most of the cases is unstructured, sentiment analysis has become a principal tool of making that data meaningful. In linguistics, Dravidian languages have a unique orthography, syntax, lexis, and morphology. Telugu is one among them with more than 90 million native people who write, read and speak in the language. There is a continuous exchange of data on the world-wide-web in the Telugu language that impacts products, reviews, news editorials, political concerns, and many more. Most of the analytics tools are developed with respect to the structure and functions of the English language and majorly spoken language across the world. Telugu language being second popular spoken language in India lacks in digital innovations like text analytics, sentiment analysis, etc. which can impact wide-range of markets.

This dissertation aims to design and implement a sentiment analyser for Telugu, a Dravidian language by building a SentiWordNet lexicon coupled with a benchmarked Telugu POS tagger. This system handles various morphological and syntactic rules in processing Telugu script which is very different from the existing sentiment analysis tools of English and other languages. A web application has been developed for extracting sentiment for given Telugu sentences. An experiment is conducted by collecting various e-newspaper articles with a manual annotation by three native speakers and comparing it with the results of the developed prototype. It resulted in an accuracy of 83.5%. It also examines a case study for detecting biased Telugu e-newspapers by comparing with a local governmnet official statement in the Telugu languages. However, results varied between articles and shows biased reviews. Also, gives an overview of adopting this framework for other Dravidian languages like Tamil, Malayalam, and Kannada. Overall, the methods presented have limitations but shows promising results in performance and scope for future research and expansion.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Digital media usage has increased in the last two decades and attracted more than 75% [1] of the population across the globe. It appears to gain the ground as a mode of communication between the individuals to officials in the digital world. Numerous messages are being exchanged using the social media, content on the blogs, e-news articles and many more on the web platform contributing to the collection of data, which keeps on increasing every day, and certainly one of the collective sources of data. This data can be helpful in business, education, policy-making, decision-making strategies and many other domains with the aid of data analytics. The Internet includes an abundance of data that can be used by individuals to assist them to decide a particular problem. Some people generally attempt to find out other people's views on the internet regarding goods, places they plan to travel to and spend a moment in, or films they plan to watch in a theater. This allows computer scientists to collect and analyze a lot of opinions and sentiments on particular subjects from these data. The method of analyzing texts of sentiments has now come to light. Instead of searching for and reading feedback to get the ultimate view it is useful for the people to access views and sentiments on a particular subject in a sensible and digital way. For example, if someone would like to purchase and does not know a particular type of phone to buy, they would surf the internet and read the product reviews. Finally, based on these assessments, a choice can be made. This manual process is a sort of opinion mining or sentiment analysis. Sentiment analysis has been advantageous to various tasks

[1] https://www.internetworldstats.com/stats.htm

such as answering systems and information extraction of Natural Language Processing (NLP). The purpose of information extraction is to mine a piece of information that corresponds to a particular subject or needs of the user. For instance, individuals have been using the internet, through blogs or other social networks, to share their ideas about subjects or problems. Some of these thoughts are positive, while others are more vicious.

In order to enable sentimental applications, this idea of online sentiment analysis spreading has created a new field in text analysis, extending the subject from traditionally informative and factual views on the text. In the last 10 years, there has been a great deal of publicity in the industries as well as in research to extract sentiment from text. Formal analysis of the sentiment tries to determine the view of people from their text. Many areas such as natural language, computer learning, and computer linguistics are covered in this subject.

The content in the digital media platform could be in different languages. There are roughly 6500 languages spoken in the world [2] and English is the most common language used in Northern America, the United Kingdom, Ireland, and Australia. It is also recognized as an official language in many other countries. Analytics tools are developed with respect to the structure and functions of the English language. As English belongs to the Indo-European language family, these analytical tools can be adapted to languages from German to Hindi, Italian to Urdu. However, there is a growing popularity of digital media output in non-Indo-European languages like Arabic, Hindi, Telugu, etc.

In India, there are 22 official languages[3] out of which Telugu is the second most spoken language in India and it belongs to Dravidian language group. This linguistic family have a unique orthography, syntax, lexis, and morphology. Telugu, Tamil, Malayalam, Kannada are most spoken Dravidian languages in the states of Telangana, Andhra Pradesh, and other southern Indian states [4] . **Telugu is hailed as the Italian of the East by Italian explorer Niccolò Da Conti** [5] because of its melodious speech and vowel ending words as Italian.

- Telugu script: ``దేషా బాషా లాండు తెలుగు లెస్సా ,శ్రీకృష్ణ దేవరాయలు "
- Transliteration: **Dēṣā bāṣā lāṇḍu telugu les'sā, Śrī krṣṇa dēvarāyalu**

- English Translation: **Telugu is the greatest of all the languages in the country said by great ruler Sri Krishnadevaraya.**

There are several factors to choose to work with the Telugu language. Firstly, because of its large audience that are now using the online resources, Telugu sentiment analysis is required to analyze and go with the best options. Second, the historical, strategy-oriented significance of this language for its country, its culture, and its legacy are exciting and challenge-oriented. A survey also states that on each day there are several thousands of messages exchanged in the Telugu language script on different online media platforms [10]. The volume of data especially in digital media is diverse in quantity and quality. It is not possible now to analyze the media manually for making intrinsic and extrinsic decisions on the data. Furthermore, the language restriction in this field begins with the source and finishes with the tools.

1.1 Motive of the Dissertation

The primary motive behind this dissertation is explained in this section. Part one outlines the research question this work is engineered to tackle and portion two gives the hypotheses that may be raised by the research questions. The main issues that are addressed in this work:

1.1.1 Research Question-1

Question What approaches can be adopted for building the Telugu Sentiment Analyzer?

- Is training a corpus and predicting using Machine learning is only approach for the Telugu language?

- How to build a Telugu SentiWordNet?

- **Description:** The motivation for this question comes from analyzing the existing language sentiment analysis techniques and could these approaches be applicable for building Telugu sentiment analyzer or not. Also, one of the easiest way to develop a sentiment analyzer is by using SentiWordNet. So, if this approach is chosen, what are the ways to develop a Telugu SentiWordNet.

1.1.2 Research Question-2

Question Does the Telugu language have enough sentiment corpus?

- Is the analysis of Telugu sentiments necessary with more open annotated information?
- What are the Telugu sentiment corpus domains and language types?
- **Description:** The reasons behind this question is the investigation of the availability of resources i.e. sentiment corpus of Telugu. Sentiment analysis in Telugu, compared to other languages like English, is relatively novel. Furthermore, classification of sentiment is a very particular problem for the domain. There is, therefore, higher knowledge of the sentiment, the more fields annotated sentiments corpora.

The following are the key hypotheses which this dissertation claims:

- **Hypothesis-1:** The research community is not given sufficient free corpus to analyze Telugu language sentiment extraction.
- **Hypothesis-2:** The Telugu language requires a variety of features and depictions, such as syntax, semantics and style, to capture sentimental concentrate.
- **Hypothesis-3:** For Telugu sentiment analysis, the application of Natural Language Processing (NLP), as the word constellation and SentiWordNet could be helpful.
- **Hypothesis-4:** The proposed algorithm can be adopted for other Dravidian languages such as Tamil, Kannada, and Malayalam.

1.2 Contributions

The strategy adopted in this dissertation to tackle and investigate as stated in the research questions and hypotheses described above lead to the following contributions. Having done an extensive literature survey to be discussed in chapter-2, it was clearly understood that, there are not many approaches available for Sentiment Analysis.

Though a few are present with a major focus on machine learning which is indirectly domain-specific. Several design decisions were made for building a universal sentiment analyzer for Telugu language. Finally, it was decided to go with Parts Of Speech (POS) based tagging as this technique results each word a POS tag. Which will be easy in classifying words in a sentence. Also, helps in identifying objective and subjective words/sentences, as objective words/sentences do not contain any sentiment and will not be useful sentiment analysis. A Telugu SentiWordNet is constructed in this dissertation to extract polarity from the given word/text as shown in Figure 1.1. This work especially focuses on different strategies opted for building Telugu SentiWordNet for better evaluation of the text. In order to judge the subjectivity of the given text, a judgement function is introduced in chapter-3 for classify the text as ***positive*** or ***negative***.

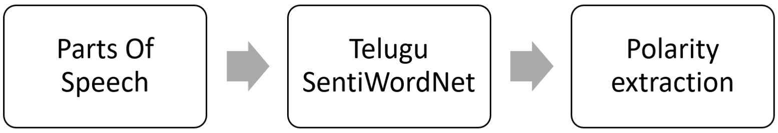


Figure 1.1: System flow overview

A web application is developed for collecting manual input from end-users to test the parsed input Telugu text. It includes a server developed on the Python Flask framework, which holds the statistical logic for calculating the polarity of the text. The end result would be a sentiment label and polarity score of the input text. This thesis demonstrates that the cross-domain method with Telugu's sentiment analysis method is possible. By using this technique, we do not have to take longer and more time to notify a new data domain type. The experiment results show the promising of using this technique with Telugu sentiment analysis. Also, a comparative study is done on detecting biases between Government official press releases versus Telugu e-newspaper articles.

1.3 Thesis structure

This section discussed the research problem in the analysis of sentiments in Telugu, which also outlines the primary goals of this dissertation, and organizes the rest of the thesis as follows.

Chapter-2: This chapter gives an overview of sentiment analysis, its uses, and needs for the system. Discusses different approaches to analyze sentiment of a text. It covers wide-range of state-of-art technologies and strategies that are used for developing a system. This dissertation is majorly focused on building sentiment analyzer for Telugu language, so in-depth literature about Telugu and its parent Dravidian languages are discussed with their technical challenges.

Chapter-3: This chapter summarises the design selection that was made for achieving the Telugu Sentiment analysis. Parts-Of-Speech tagging strategy, process of building a Telugu SentiWordNet using different approaches. A critical analysis of the developed Telugu SentiWordNet. The judgment function for extracting sentiment polarity.

Chapter-4: This chapter provides an overview of technical implementation with the chosen technologies and their logical dataflow. Presents technical architecture of the proposed prototype. Also calculates the time complexity of the algorithm implemented in terms of Big-O notation.

Chapter-5: This chapter evaluates the implemented prototype by collecting e-newspaper data from different Telugu e-newspaper sites on the web. Manual annotation is considered as ground-truth results and compared with system results. Also, gives the time performance of the program execution.

Chapter-6: The final dissertation remarks are concluded. This chapter ends with recommended directions for future research and improvements.

Chapter 2

Background and Literature review

This chapter introduces to the Dravidian language family group especially Telugu language and expresses motivation for the digital innovation for minor languages like the Telugu language. Links to the persuade research context by logically connecting to NLP, Text Analytics and Sentiment analysis. And the finally strong motivation of this dissertation.

2.1 Overview of Research context

2.1.1 Dravidian Languages

The Dravidian languages are one of the linguistic family with approximately 215 million native speakers in South Asia and the rest of the world. This is also the fifth-largest linguistic family in the globe. It dates back from 4500years with no clues how it is originated or adopted by the Asians especially in South India and Pakistan [6]. Most of the linguists consider that the Dravidian language group is entirely unrelated to other languages in the world. This linguistic family has been divided into 73 different dialects spoken in South India, Pakistan, Nepal, Sri Lanka, Afghanistan, Maldives, Burma, Mauritius, and few other Asian countries. The Dravidian languages are predominantly spoken in Southern states of India whereas Indo-Aryan linguistic family is observed in rest of India. Historians believed that, initially, Dravidian languages are native for the Indian subcontinent and Aryan immigrants from the west brought the Indo-Aryan dialects to the Northern Indian and pushed Dravidians to the south part of the country

[7]. Dravidian languages are typically divided into communities depending mainly on their geographical distribution such as Central, Southern, Northern, and South-Central as shown in Figure 2.1 adopted from [8].

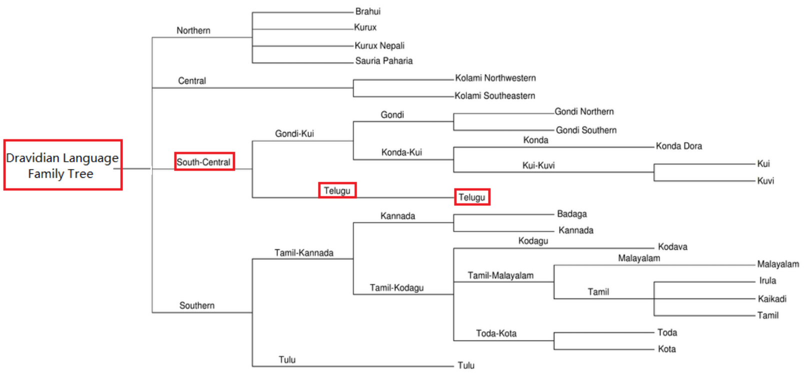


Figure 2.1: Dravidian language family Tree

Language Status

Out of 22 official languages and 14 regional languages in India, the four major languages from Dravidian language group are Tamil, Telugu, Malayalam, and Kannada which are officially recognized by the Constitution of India. These languages have an extraordinary wealth of literature and used for administrative, media, education and business. In 20th-century India, these four languages have also impacted the country and people in social, political, technical, and economic changes.

Structure

- Vowels: Most Dravidian languages can be lengthy or brief with five or more vowels. The length of the vowel differs in the significance of the same phrases.
- Consonants: Apical tongue consonants with a tongue tip are produced on the mouth's roof, while a retroflex consonant is produced curled with the tongue so that its underside contacts the mouth's roof. Has a variety of nasal consonants.
- Grammar: The family of Dravidian languages falls under agglutinative which means in order to derive a grammatical relation there is a need to add suffixes to

stems. These are joined one by one and sometimes lead to very lengthy phrases. Most common characteristics of agglutinative languages are to use postpositions rather than prepositions to mark grammatical relations.

2.1.2 Literary languages

As stated above, the four literary languages of the Dravidian group are Tamil, Malayalam, Kannada, and Telugu.

Tamil

Tamil is a medieval language in India with heritage literature, it is considered as official languages in Singapore, Sri Lanka and Tamil Nadu, Pondicherry states of India. This language partially influenced by Sanskrit, the Devanagari script which is an ancient script in the world. Introduced to the world in publications during British-India rule [9].

Malayalam

Malayalam is the official language of the Indian state Kerala, it has many lexicon borrowings from Tamil and Sanskrit. The pronunciation is very melodious and very agglutinative. In Malayalam, a phrase gives semanticized and syntactic relationships between phrases and nouns and other components[10].

Kannada

Kannada is the official language in the Karnataka state of India, it shares its lexicon and script with the Telugu language. From the letters Kadamba and CāLukya, descendants of Brahmi used between the 5th and the 7th millennia AD, the Kannada alphabets were produced [11].

Telugu

Telugu is the official language of twin Indian states Telangana and Andhra Pradesh, it is the second-largest spoken language in India. Though Telugu is of the Dravidian group, it has great roots from Sanskrit and Urdu. Telugu was called as the Italian of the East by Italian explorer Niccolò Da Conti[12] because of its melodious dialect and vowel ending structure alike Italian, a popular Indo-European language. This research is majorly focused on building Sentiment Analyser for the Telugu language.

Overall, these four languages are strong pillars of ancient Dravidian language family and having official status in various countries but, in the research context, it has not seen a ground breakthrough as other major languages. In this dissertation, a lexicon-based sentiment analyzer proposed for extracting sentiment from the Telugu texts and believes the same framework can be adopted for other Dravidian languages too.

2.2 Natural language Processing and Text analytics

In the modern era of internet technology, data generation is at a greater pace by producing numerous quantities of data online and offline. Virtual activities such as website texts, articles, posts in blogs and social media are engendering unstructured textual data. Natural Language Processing (NLP) is the theory-motivated range of software methods for assessment and classification of natural language. For natural language processing, there is no conventional meaning, it entirely relies on the research and development context [13]. The most commonly researched tasks in NLP are syntax, semantics, speech, analytics etc. This subject has a strong view on sentiment analysis.

2.2.1 Sentiment Analysis

The word 'sentiment' must first be decided upon before a sentiment analysis is discussed with regard to any language. As per the Oxford online dictionary[1], the word "sentiment" means ***"A view or opinion that is held or expressed or feeling."***. This can be defined in many other ways depending upon the research context[14]. "What other people think", for most of the people in the decision-making process is always one important piece of information[15]. Linguistic researchers have given huge attention to the sentiment analysis from the 20th century and progressed briskly for analyzing the text from a raw data corpus or database. Worldwide, texts can be divided into two primary kinds of data: facts and opinions. Facts are objective terms regarding

[1] https://www.lexico.com/en/definition/sentiment

organizations, places, events, and their characteristics. Opinions are typically subjective phrases that define the emotions, assessments or emotions of individuals towards organizations, etc. As per the literature from [16],[17],[18], sentiment is broadly diving into two subcategories, they are:

- Objectivity: In general, objective sentences are contemplating and portraying facts, not affected by personal sentiments or opinions.

  Example, *'Trinity College Dublin is in Ireland"*

- Subjectivity: Subjective sentences are based on individual feelings or tastes, or opinions.

  Example: *"the movie that released yesterday is not good and may impact young people"*

Sentiment analysis is a subfield in the Natural Language Processing and Computational linguistics [17]. It is a way to understand people's feelings in other terms sentiment or opinion towards a particular subject, this also known as opinion mining or subjectivity analysis. In general, the sentiment of a subject can be explicit or implicit.

In case of an Explicit sentence, the sentiment is openly understandable for example జాన్ మంచి అబ్బాయి */ Jān mañci abbāyi/ John is a good boy*[2]

In the case of Implicit sentence, the sentiment is unclear clear and completely depends on the context of the pre-text and post-text. For example, జాన్ చెడ్డ అబ్బాయి కాదు మంచివాడు కూడా కాదు */Jān ceḍḍa abbāyi kādu mañcivāḍu kūḍā kādu /John is not a bad boy but also not good .*

There are many different ways to extract sentiment from a text/sentence. One such method is to perform Objectivity analysis and Subjectivity analysis, this is a two-step process and most commonly used approach for mining sentiment from the text. In the initial step, the given text is identified as objective or subjective, objective words or sentences are discarded from extracting sentiment as they don't carry any emotion or opinion implicitly or explicitly. In the next step i.e. if found subjective, then an analysis is performed to know the text has a ***positive*** or ***negative*** opinion

[2] In this dissertation, Telugu sentence is represented in three variants: Telugu sentence / Transliteration / English translation as the aim of this work is perform Sentiment analysis for Telugu language.

[19]. The subjectivity analysis is an important step to know the sentiment of the text, this becomes challenging if another parameter that is ***neutral*** is added to the list. This research is focused on this approach as it aims to perform sentiment analysis for Telugu language and lack of other resources to support the Telugu language technically.

Machine Learning-based approach is a successful classification model for carrying out sentiment analysis. In this approach, firstly a model needs to train and then predict based on the trained corpus [20]. It is the best way to extract sentiment for domain-specific problems. For instance, sentiment analysis for movie reviews or product reviews.

Another form of sentiment analysis deals with the discovery of the sentiment target. Much of the work accomplished in the area of sentiment analysis is concerned with discovering feelings about a particular subject or objective, such as customer feedback on a film or product [21]. It is simple to assess the subject in such reviews because one assumes that the evaluation speaks about a certain product. In contrast, in the event of an unidentified goal, using feature-based sentiment analysis would be more challenging. Thus, a review has to be done first to identify what characteristics a customer has recorded by using extraction techniques.

2.2.2 Sentiment Analysis Challenges

Sentiment analysis or classification is usually regarded as a unique field in natural language processing's text classification. Classification of topics depends on keywords, but in general, this formulation does not function very well in the event of an evaluation of sentiments[22]. Other sentimental analytics problems arise from the complexity of the issue. Sometimes the adverse sentiment may be conveyed in a phrase without using evident adverse phrases. In addition, there is a fine line between the objective and subjective label of a phrase.

Identification of the opinion holder in the sentence is the toughest tasks to judge the subjectivity of the sentence. The analysis of the feeling relies greatly on the data's domain. The phrases often have favourable feelings in a particular domain, but in a separate domain, they have a distinct feeling of polarity. Subsequently, a number of other styles of writing like humor, sarcasm or negated sentences might present more practical problems to the sentiment analysis.

2.2.3 Usage of Sentiment Analysis

Sentiment analysis is a practical field of NLP with enormous applications, this tool can be helpful in estimating how users are feeling about the movies or products, etc. in the digital world. Businesses recognize the significance of the Internet to gather views and recommendations on the products and facilities of customers in a market place. Typical consumers often invest some effort browsing the internet to determine others views, whereas companies usually need an automated system to assist them to find the feelings and opinions of their products and services. It is more important for companies to use a tool that can acquire and evaluate customer feedback to know the ultimate sense. This tool can give the emotions and thoughts of the clients to enhance their products and facilities.

Social media usage has increased in the last two decades and attracted more than 75% of the population across the globe [23]. It appears to gain the ground as a mode of communication between the individuals to officials in the digital world. A lot of emails are shared via the digital media which contributes to the daily growing information set, definitely one of the personal information outlets. Through information analyses, this information can be useful in business, schooling, decision-making, and many other areas.[24].

2.2.4 Top Benefits of Sentiment Analysis

- Public view on a Political product.
- Predict Movie reviews.
- To know the launched product is successful or not.
- To know how the reputation of the brand develops over time.
- React faster to customer messages and changes.
- Quantify otherwise qualitative information and etc.

There are many more benefits that sentiment analysis would leverage for different sectors[25].

2.3 Methodologies of English language Sentiment Analysis

The sentiment analysis was developed and tested by a wide spectrum of methods and technologies. Most of these methods are based on the prevailing language of science i.e. the English language. This did not prevent computer scientists from designing methods for other languages such as Chinese, Korean, Japanese and Arabic. In traditional sentiment analysis, most of the implementation is categorized as supervised and unsupervised learning methods.

2.3.1 Supervised learning-based methods

The supervised learning approach is a strategy for Machine Learning (ML). This technique converts annotated information to function vectors and uses ML classification to produce a mixture of certain characteristics generating a particular category [26]. A training set and classification test set is employed in machine learning methods. The training set includes vectors for entry features and their category marks. Several machine learning techniques are used to classify reviews like Naive Bayes(NB), Max. Entropy(ME), and Support Vector Machines(SVM). The term presence, term frequency, negation, n-grams and part-of-speech are some of the characteristics which would be used for sentiment classification.

Pedro et al. [27] have discovered that Naive Bayes performs well for certain extremely based characteristics issues. This is unexpected because Naive Bayes' fundamental hypothesis is that the characteristics are autonomous. In a new model, Niu et.al [28] introduced effective approaches for selecting, weighting and classifying function. This model is based on the algorithm of Bayesian and claims that results are finest while comparing to other existing systems [29] [30]. However, term weighting has been productive research and development for the recovery of data since the 1970s. Salton et.al [31] discussed three considerations for word weighting which should be regarded. Term frequency, reverse frequency and normalization are the three elements. On the basis of the above hypothesis, many techniques for the extraction of data were suggested for word weighting. Soon Zhi-Hong [32] have worked on developing supervised learning-based sentiment analysis and quoted that documents representation is a key component of several tasks, as well as the analysis of sentiment. The Vector Space Model (VSM) is widely used for document representation. By VSM, a word (or feature) space matrix is displayed on each document. The weight (or values) of each vector variable are the main contributors to the VSM depiction of the documents which yielded in best results.

The majority of current text representation algorithms are focused on design of the syntactic background of phrases but overlook the sentiment of the text. This issue is highlighted by Duyu et.al [33], in this study authors considered learning of sentiment-specific term integration, encoding feeling data in an ongoing term depiction. In particular, three neural networks were created to efficiently integrate text feeling polarity supervision. An experiment learning-based to the unified model combining syntactic context phrases with emotion data produced the greatest efficiency of 81%. Convention of an affiliation scoring with a small set of positive and negative words to calculate corpus-based semantic values was proposed by Kennedy et.al [34]. It demonstrates that the extension of the term counting process by contextual value shifters increases identification precision. Also, uses a Machine Learning algorithm, Support Vector Machines to evaluate the model.

2.3.2 Unsupervised based-learning methods

The lexicon-based technique is the most transparent way of carrying out unsupervised sentiment analysis. These techniques use a predefined lexicon of sentiments to determine the overall polarity of the sentiment of a certain text. These are divided into two classifications:

- Human annotator's methods
- Dictionary-based methods

Human annotator's methods

In this category, a group of people who are experts in their specific or targeted language are used to manually label a series of words for creating the sentiment lexicon. General Inquirer System by Stone et.al [35] is a perfect model for human-annotated sentiment lexicon in association with IBM. In the initial days of content analysis, this system was very efficient and was adopted by many other industries and individuals for performing content analysis [36]. Later, Multi-perspective Question Answering (MPQA) corpus by Wilson et.al [37] was created in which the subjective phrases, plus all the expressive subjective features, are immediate subjective statements with non-neutral emotional intensity. To advance this system, Lingjia et.al [38] presented a system to add MPQA corpus annotations of entities and events. The new system has already annotated a subset of MPQA. The corpus is a vital source for the development of sentiment analysis technologies on the level of entities/events to enable NLP apps like automatic answering questions.

Dictionary-based methods

The dictionary strategy utilizes a predefined dictionary of phrases with a particular sentiment polarity connected with each phrase. Each of the texts is separated into one-word tokens, which maintain frequency counts it is called as a 'word bag' model. The sentiment is then calculated by combining phrases with records in the dictionary used. A particular language dictionary is used for research in order to analyze sentiments. While some specialty dictionaries are created, a current lexicon, such as the General Inquirer dictionary, is prevalent.

Rocksteady is an affect assessment system created by Ahmad et.al [39],[40] from Trinity College Dublin uses a dictionary-based sentiment analysis strategy. It offers a library of general and domain-specific dictionaries that enables the user to choose a variety of basic and specialized dictionaries for the analytical process. Ahmad et al. [41] also uses the Rocksteady tool to evaluate media papers in the run-up to the Irish elections in 2011 to forecast the result. The research produced successful outcomes that were extremely near to the ultimate voting results.

There are different other tools available which are handier in performing dictionary-based sentiment analysis. SentiWordNet, LIWC etc.

SentiWordNet

SentiWordNet[3] is a commonly used tool in opinion mining that is centered on the WordNet English dictionary. This lexical dictionary groups into synonymous collec-

[3] http://sentiwordnet.isti.cnr.it/

tions, synsets, adjectives, nouns, phrases, and other grammatical categories. SentiWordNet uses a WordNet dictionary synset to identify the phrase sentiment: positive, negative and neutral. The values of [0, 1] and up to 1 shall be acquired by means of a semi-supervised ML technique. For instance, assume a certain synset was obtained from a tweet, s= [horrible, evil, dreadful]. SentiWordNet will then offer 0.0 for positive, 0.850 for negative, and 0.150 for neutral. A labeled lexicon dictionary is used for SentiWordNet evaluation.

In this dissertation, the concept of SentiWordNet is used for developing Telugu SentiWordNet. The reason behind this decision is Telugu language being morphologically different to others can leverage the existing English SentiWordNet dictionary in creating a Telugu SentiWordNet. This can be achieved by collecting an English-Telugu pair dictionary and map the English words with the English SentiWordNet.

Linguistic Inquiry and Word Count

Linguistic Inquiry and Word Count (LIWC) is a text analysis tool that determines verbal, behavioral and functional aspects of a specified text on the basis of the word and classification dictionary. LIWC offers other classes of feeling in relation to the detection of beneficial and bad effects in a specified document. The term “agreement” belongs, for instance, to the previous classifications of words: consent, emotional, positive effect and negative.

2.3.3 Sentiment Analysis Common Features

- Term Frequency is the assessment of how often in a document a particular word is listed. The word presence indicates the existence of the word in the document in a linear way. This is highlighted in traditional IR systems.

- The POS scheme reduces the uncertainty of the word tagging, it helps to boost the trust of the NLP system in the true significance of a word with its POS label. In the case of more morphological languages such as **Telugu**, this will help considerably.

2.4 Telugu Language

The Telugu lexicon demonstrates an overall Sanskrit impact which is at least 1 500 years old, but there is proof that an old Sanskrit impact is evident. Nannaya's[4] reworking of the Mahābhārata in Telugu (along) recreated in the 1000-1100 CE era and ruled over the holy Sanskrit dialect. The Sanskrit Actamas were adopted by Telugu.

2.4.1 Telugu Language Structure

Vowel

- Telugu has two patterns of 5 vowels, i.e., tones which change the significance of the phrase.
- There is a brief and a long vowel in each pattern, the vowel size differentiates between the same phrases.
- Two diphthongs /li/ and /lu/ are available.

Consonants

Telugu's consonant system is comparable to other Dravidian languages and the features are:

- There is a comparison between simple and aspiring pauses, voiceless and voiceless. For instance, $/k - k^h, g - g^h/$
- Consonant groupings are allowed in initial and medial positions, for the most part. There are no end-position clusters.

Word order

The word order for the Telugu language is Subject-Object-Verb (SOV), it is the most prevalent form in natural languages with a word order preference.

Example **Telugu sentence** మేరీ పాట పాడుతోంది

Translation: Mary is singing song

[4] https://en.wikipedia.org/wiki/Nannayya

Table 2.1: Word order of the Telugu language

Subject-Object-Verb Order			
Words	మేరీ	పాట	పాడుతోంది
Transliteration	Mērī	pāṭa	pāḍutōndi
Gloss	Mary	song	singing
Parts	Subject	Object	Verb

As said in the previous section, Telugu is a member of the Dravidian language family's south-central division with about 90 million people speak it in the first language and 5 million people speak it in India in the second language. It is an administrative language for two states which includes wide-range of industries and markets. In the internet era, the data is being produced in quintillions and mostly unstructured. This data can be from any sources such as social media, digital media, and content in the webpages, short messages and many more. NLP has given some beneficial tools for analyzing this raw data and extract some important information that can impact improvements in the services or products or any sector. Though these tools are focused and structured according to English or majorly spoken languages in the world. Also, an important thing to note here is these systems would not work for the Telugu language due to multi issues such as morphological differences, syntactic rules, word order, etc.The word order of the Telugu language is Subject-Object-Verb (SOV) whereas for English it is Subject-Verb-Object (SVO).

2.4.2 Challenges of NLP for Telugu language

- Three kinds of affixes can be used for the terms of Telugu: prefixes, infixes, and suffixes. Affixes can be a single word lengthy or a various word combination.

- The amount of ambiguity of Telugu morphemes is noteworthy in relation to their complicated nature.

- It is not simple to determine whether a word is an affix or a portion of the stem. Speech taggers, morphology analyzers, identity name recognition, and syntactic parsing, all influence the NLP-tools that cope with Telugu.

- Lexical ambiguity (word level) – various meanings of words.
- Syntactical ambiguous-various methods to parse the phrase.
- Qualitative information — the phrase context may affect the phrase.
- There has been very less research on NLP for Telugu language.

2.4.3 Methodologies of Indian languages Sentiment Analysis

Indian languages have seen very less attention in terms of the research context, one of the major reason can be English is the official language in India. However, the Government of India has enforced and allocated funding for saving the regional languages. In recent times, the premier universities in India such as IIT's, IIIT's and state-owned universities have established linguistic research labs. In the initial stages, the researchers are focused on learning the morphology of the Indo-Aryan language family such as Hindi, the most spoken language by the citizens (approximately 53%) of the country. As previously discussed in section 2.1.1, there are two groups of language families exist in India. One is Indo-Aryan and other is Dravidian language family. Most of the Indian languages have a similar dialect and especially same word order i.e. Subject-Object-Verb structure. So it is decided to follow the NLP principles of existing Indian languages such as Hindi, Bengali, Odia for buidling Telugu sentiment analyser.

A machine learning-based NLP system for Indian languages was introduced by Mallamma et.al [42] which uses Statistical Machine Translation (SMT), it is a computing paradigm in which texts are created using statistical designs which analyzes of bilingual text corpora are used in the assessment. The statistical method contradicts both machine translation rules and machine translation examples. For Indian languages POS tagging was researched by [43], they provided a tag set with a morphologically productive and agglutinative language for the Indian language Malayalam, a relatively free order. The article discusses how to use an SVM tool that uses vector makers and a tagger, creed with the Hidden Markov model, to tag the Indian language Malayalam. The SVM tool gave 87.5% precision and the TnT Tagger gave approximately 80% precision. In a study by Chandramma et.al [44] has presented a cross-language transfer model of learning for Telugu using Kannada resources. In this article, the authors introduced a morphology assessor and build big corpus for both Kannada and Telugu. The morphological analyzer for other Indian languages can be expanded to these models. Experiments are intended to assess how well both monolingual and cross-language Tagger is performing a suggested POS Tagger model. The research demonstrated that the suggested strategy is efficient and much quicker than the state-of-the-art POS tagger model.

To counter the dictionary-based POS tagging, Rama et.al [45] used a voting algorithm, they tried to enhance the precision of the current Telugu POS taggers trained with an annotated corpus of 12,000 phrases. Three taggers for Telugu POS are proposed: (i) Brill Tagger, (ii) Rule-based (iii) EntroPy POS Maximum taggers yielded precision 98.016% 92.146% and 87.818%, respectively, and created with an exactness of 98.016%.

2.5 Summary

This chapter introduced the Dravidian languages in the research context. Also, presented various approaches for developing sentiment analyzers with in-depth literature and analysis. Elucidated importance of having a sentiment analyzer in the digital platform. Gave an overview of the Telugu language which the primary focus in this dissertation. The limitations of NLP for adapting to the Telugu languages were stated with practical scenarios. Finally, justified the research question to built a Telugu corpus for sentiment analysis.

Chapter 3

Method

3.1 Introduction

This chapter is intended to provide an outline of the steps adopted to build the proposed system. Specifically, the scope of the work undergone is divided into different phases: developing a Telugu Parts-Of-Speech tagger, Lexicon for Telugu Sentiment extraction, Data acquisition, and analysis of data. Thus, in this chapter, each section gives an outline of the methodology used in the research stage.

Figure 3.1 shows the proposed system architecture and its components which are essential in achieving the goal of developing universal Telugu sentiment extractor.

3.2 Design

Considering the proven best-solutions for the English language, initially thought of using a similar design mechanism for the Telugu language, but Telugu is a morphologically different language than English hence, these principles were not applicable to Telugu. So, a new design is proposed and implemented in this thesis. The proposed design have different components, which includes data collectors, language detector, Telugu language Parts Of Speech (POS) taggers, data analysis, filters, Telugu SentiWordNet, sentiment extraction, analyzer, noise filter, sentiment extraction engine and finally a Graphical User Interface (GUI). Figure 3.1 depicts the flow of events of the designed model.

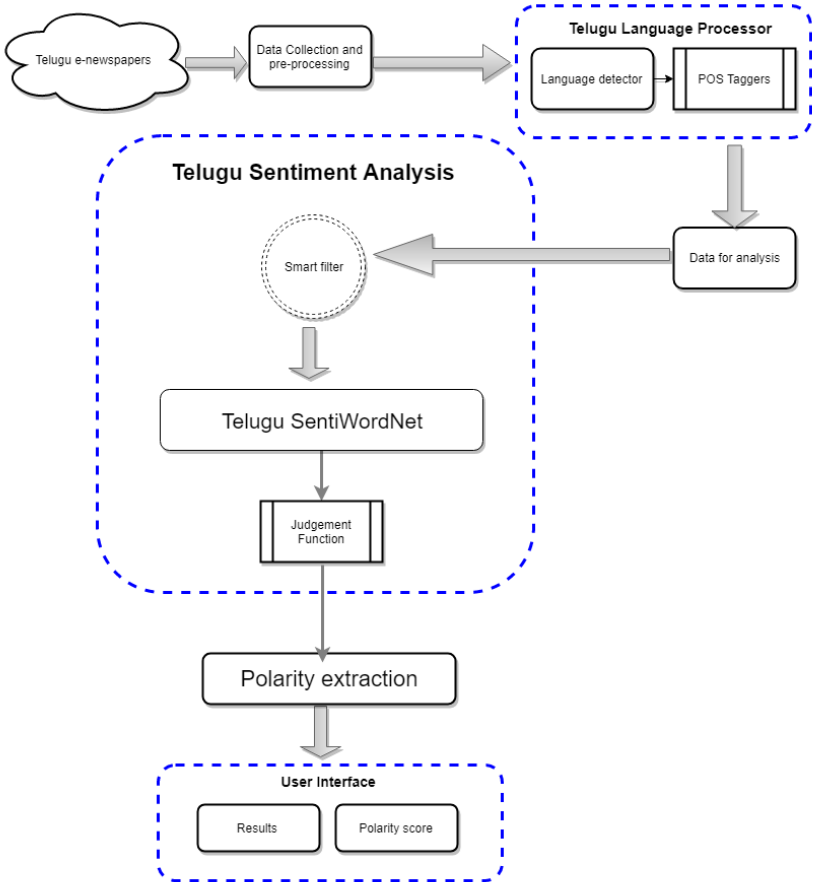


Figure 3.1: System design overview

3.3 Telugu Language Processor

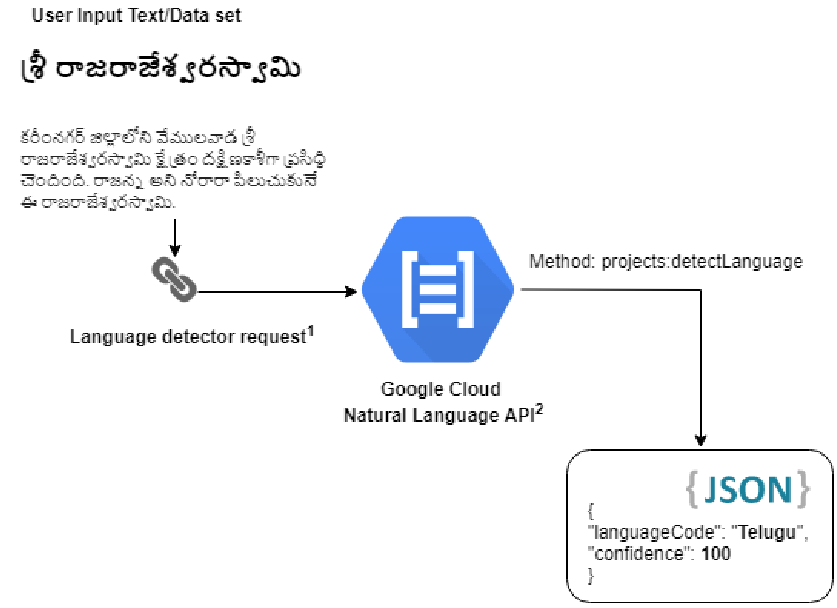
This is an important component of the designed system, the computation processing of this competent is key for extracting sentiment of given words or sentences or paragraphs. This processing unit includes two core modules: 1. Language detector and 2. POS Taggers.

3.3.1 Language Detector

Language detection is a key process to know whether the inputted data is in the format of expected data or language. Natural Language Processing libraries have evolved to produce these results, many researchers have proposed various techniques [46][47][48][49]. However, the most popular language detector or translator is the

Google Inc.'s Cloud translator is supported for a broad range of dialects by the Translation API identification engine. In this thesis, the Google API services with legal permission are used to identify the Telugu script and process to the next level to get efficient results.

Figure 3.2 shows the overall flow for the language detection, firstly, an input text is parsed as an HTTP request[1] to the cloud natural language API[2], the "detectLanguage" service will aggregate the data and responds with the language code. In this case the expected language code in ISO-639-1 format, i.e for Telugu it is "te".



శ్రీ రాజరాజేశ్వరస్వామి

కరీంనగర్ జిల్లాలోని వేములవాడ శ్రీ
రాజరాజేశ్వరస్వామి క్షేత్రం దక్షిణకాశీగా ప్రసిద్ధి
చెందింది. రాజన్న అని నోరారా పిలుచుకునే
ఈ రాజరాజేశ్వరస్వామి.

```
{
"languageCode": "Telugu",
"confidence": 100
}
```

Figure 3.2: Telugu language detection overview

3.3.2 POS Taggers

As stated in the prior discussion (Chapter-2), there are many approaches to design a POS tagging system, it is proven by many researchers that Machine learning approach is an efficient way to train and test the data. But, for a universal POS tagging system, this approach may fail as the data trained can be of domain-specific. To tackle this issue, it was decided to go with Trigrams'n' Tags (TnT) based Parts-Of-Speech tagger

[1] https://translation.googleapis.com/language/translate/v2/detect accessed on 25th-June-2019

[2] https://cloud.google.com/translate/docs/reference/rest/v2/detect accessed on 28th-June-2019

considering the utmost Telugu vocabularies list. Figure 3.3 shows the overall flow of the POS tagging system. For any natural language processing, (i).Morphological analyser and (ii). POS Tagger are two tools which are most essential. The next thing to note here is that POS tagging is also important to comprehend the target language in machine translation.

Trigrams‘n’ Tags (TnT) based Parts-Of-Speech tagger

It is a statistical tagger that works on second-order Markov models.

- It is a powerful parts-of-speech tagger that is trained in languages and tags. It can be used in different languages
- For parameter generation, the component trains on tagged corpora. It includes various filtering techniques and unidentified language processing.
- For filtering, linear interpolation is used, the weights are determined by the omitted interpolation.

The tagger model consists of the probabilities for transition and emission. Where tags are symbolized in-state/transition of model and words are characterized in output/emission. The probabilities of transition rely on tag combinations, i.e. statements. The probability of emissions relies primarily on the previous classification. The tag sequence of given word sequences is chosen by computing [44].

$$argmax(t_1 \dots t_n) \left[\prod_{i=1}^{n} P\left(t_i | t_{i-1}, t_{i-2}\right) P\left(w_i | t_i\right) \right] - - - (1)$$

In the above equation, the corresponding POS tags are $t_n \dots t_{\mathcal{K}}$, t_{-1} is auxiliary tags, t_0, is beginning of sequence markers and $t_{\mathcal{X}+1}$ is ending of sequence markers. Where $w_n \dots \dot{w}_{\mathcal{X}}$ is the word sequence of length X. The probabilities of transition and emission are calculated from a tagged dataset. First, computes the Maximum Likelihoods (ML), probabilities $\hat{\mathcal{P}}$ from the derivative comparative frequencies.

- Unigrams : $\hat{\mathcal{P}}\left(t_1\right) = \mathcal{R}\left(t_1\right) / _{\mathcal{T}}$ —(2)
- Bigrams : $\hat{\mathcal{P}}\left(k_3 | k_2\right) = ^{\mathcal{R}(k_2, k_3)} / _{\mathcal{R}(k_2)}$ —(3)

- Trigrams : $\hat{\mathcal{P}}\left(t_3|t_1,t_2\right) =^{\mathcal{R}(t_1,t_2,t_3)} /_{\mathcal{R}\left(t_1,\hat{t_2}\right)}$—(4)
- Lexical: $\hat{\mathcal{P}}\left(w_3|w_2\right) = \mathcal{R}\left(w_3,t_3\right) /_{\mathcal{R}(t_3)}$ —(5)

$\mathcal{T}$ T is the complete tokens in the data set being educated. It also thinks or states that the probability of ML is null if the corresponding nominators and denominators are null. Context frequencies are also simplified and lexical frequency sentences not included in the lexicon are treated, as described obviously in the next chapter (Normalization).

Normalization

In contrast to popular belief, trigram probabilities can not be generated straight from the dataset because of the scarce problem of information. As a result, there is no enough phenomenon to calculate the probability reliably for each trigram. In addition, placing a probability to null has unwanted effects because it may never happen in the dataset to have the corresponding trigram. Thus, the probability for the entire series is put to null.

A linear interpolation of unigrams eq(1), bigrams eq(2), and trigrams eq(3) represents the normalization parameter that provides the highest result in Trigrams'n' Tags (TnT). So we calculate the probability of a trigram accordingly [44]

$\hat{\mathcal{P}}\left(t_3|t_1,t_2\right) = \beta_1\hat{\mathcal{P}}\left(t_3\right) + \beta_2\hat{\mathcal{P}}\left(t_3|w_2\right) + \beta_3\hat{\mathcal{P}}\left(t_3|t_1,t_2\right)$ —(6)

where $\mathcal{P}$ represent probability distributions. Since, $\hat{\mathcal{P}}$ are Maximum Likelihood (ML) approximations of the probabilities, and β_1+ β_2+β_3= 1.

The value of β in this work doesn't rely on the specific trigram because we use linear interpolation, which is context-independent. Due to sparse data problem, one cannot calculate a distinct number of βs for each trigram. Therefore, we cluster trigram by frequencies and calculate clustered βs pairs. No previous research has regarded frequency classification depending on the understanding for linear interpolation in POS tagging.

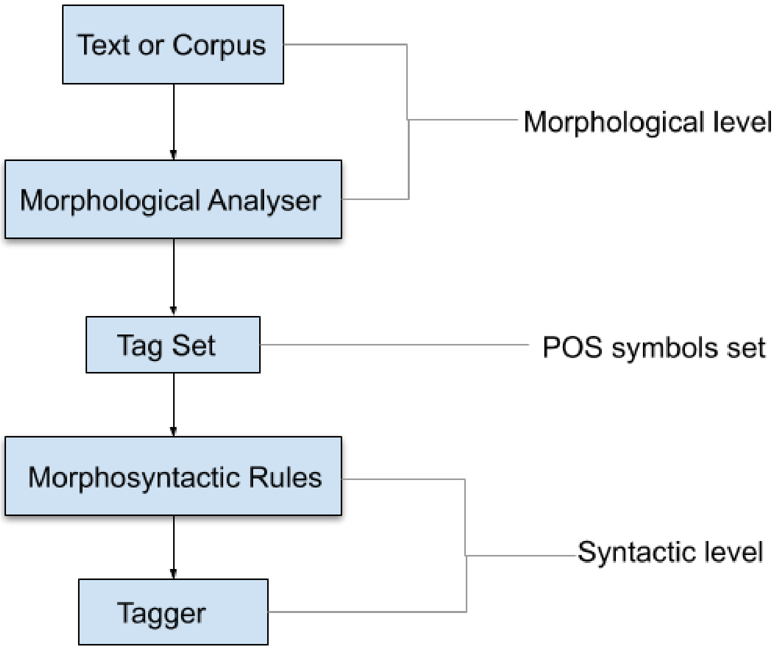


Figure 3.3: Telugu language Parts-Of-Speech flow overview

3.4 Text or Corpus

This work majorly focuses on building a sentiment analysis for Dravidian languages especially Telugu, data was collected from various trusted sources. Predominantly form Sketch Engine (Elexis) an European Union funded project[3], it is a Lexical Computing corpus manager and text analysis software. It maintains multi-Language corpora, multi-billion term lists of documents, provides reference information for all characteristics of Sketch Engine. The corpora are marked and annotated to be prepared for complicated analyses of sentences and language constructions. Parallel and multilingual corpora are also accessible. This projects extends all its glorious features to the Telugu language too. The Telugu language word list was collected from this repository.

[3] https://www.sketchengine.eu/elexis/

3.5 Morphological Analyzer

A Morphological Analyzer (MA) is a program that compiles and analyzes real linguistic phrases and their morpho-syntactic components into their origins and characteristics[50]. This research utilizes the existing Morphological Analyzer developed by G.Umameshewar Rao[51] and [50] as these algorithms gained popularity and effective in analyzation for the Telugu language.

3.6 Tag set

Tag set is a list of parts-of-speech types which are added to each words depending upon the morphological rules of the language. The process of POS tagging of a word is vital in sentiment extraction. There are many different approaches to tag a word as given in [52] [53] [54]. However defining a tag set makes the system efficient. In English, there are many different types of POS tags like: Noun, Pronoun, Verb Adjective etc. with specific morphological rules. But when it comes to Telugu syntactically different and same rules as English would not help in achieving a right POS-tag.

All the effort was given to the majorly spoken languages across the globe, Indian languages have seen such systems in recent times. One of such work is by [1] where a large Annotation of Corpora (AnnCorra) was built as a part of Indian Language Machine Translation (ILMT) project. The ILMT initiative is organized by several universities to constitute a consortium and to develop MT schemes for several Indian language exchanges. AnnCorra could be implemented at several levels like POS, phrase/clause level, dependency level, etc. The basic step towards constructing an annotated corpus is part of speech tagging. The next level is the Chunk tagging. The objective of ILMT project is to come up with a standard POS Tagging and Chunking Scheme for Indian Annotation Languages (AnnCorra) and develop codes that are exhaustive for annotation tasks for Indian languages.

Assumptions:

- The tags for all Dravidian languages should be prevalent.
- It should be comprehensive/ complete.
- Consistency in annotation.

- POS tags are NOT substitutions for morph analyzers which means the POS label must be modeled on the word's 'category ' and the characteristics can be obtained from the morph analyzer.

For example, take a Telugu word `రాముడు'
Transliteration: 'rAmudA'
Translation/meaning: Does it mean 'Ram'?

This word contains the following information: <category (noun)+grammatical features(masculine, singular) + question>.

The word by itself is a bundle of linguistic information. In this research for the purpose of POS tag set, it was decided to use AnnCorra approach due to its ease-of-use feature and benchmarked by several researchers and organizations across the world. The POS tagging and chunking standards for Indian languages include 26 POS tags (Appendix Table-2) and 11 chunk tags (Appendix Table-3).

3.7 Morpho-Syntactic Rules

Morpho-Syntactic rules are required to offer each word the appropriate POS tag. Based on the situation where the word or parts of phrases happen in a phrase, these rules are created.

For instance take a phrase, in English: John's hand
In Telugu: జాన్ చేయు
Transliteration: Jān cēyi
Here చేయి (cēyi) has two meanings (i). Hand, (ii). Do it
In that case we need to have the thematic rules.

3.8 Tagger

This section consists of two major modules, (i) Tokenization and (ii) POS tagging. Initially, the requested Telugu script would be taken and tokenization is applied on each sentence which would split each word from the sentence then it is passed to the POS tagging system where the respective POS tag is appended with each word.

3.8.1 Tokenization

Tokenization is a critical task in every text analysis model. All words, numbers, characters, etc. are actually separated from a particular text and the recognized words, numbers, and other elements are known as tokens[55]. This method also evaluates the frequency significance of all tokens in the entry files together with the token sequence. This activity yields in improvising the performance of the system as there is no need to process the word which is redundant, in this case simply append the same POS tag to the redundant word. To identify each sentence "<s>" tags are used at the beginning of the sentence and end of the sentence. Example of Tokenization:

In English: Telangana and Andhra Pradesh are twin states for Telugu people

Transliteration: Telaṅgāṇa mariyu āndhrapradēś Telugu prajalaku jaṇṭa rāṣṭrālu

Before tokenization: నాకుఇవ్వు తెలంగాణ మరియు ఆంధ్రప్రదేశ్ తెలుగు ప్రజలకు జంట రాష్ట్రాలు

After tokenization: <s> నాకుఇవ్వు, తెలంగాణ , మరియు , ఆంధ్రప్రదేశ్ , తెలుగు , ప్రజలకు , జంట , రాష్ట్రాలు <s>

3.8.2 POS Tagging

POS Tagging processes a sequence of words, and attaches a part of speech tag to each word such as nouns, verbs, adjectives, and adverbs, etc., it is a rigors process to handle many morphological rules, syntactic rules and most importantly strong language skills. Much research was conducted in automating the POS tagging, which makes Natural Language Processing easy for adopting in various text or speech analysis. As discussed earlier, Trigrams'n'Tags (TnT) tagger and Hidden Markov Models (HMM) based approach is used in this research for POS tagging. TnT tagger is a very effective, language- and tag-trainable, statistical part-of-speech tagger. The component for parameter generation trains on tagged corpora. The scheme contains multiple filtering and unfamiliar phrases processing techniques. POS tagging system for this research was adopted from [56] whose results are proficient and significantly faster than the existing monolingual tools.

Input Telugu word/sentence from Tokenization: నాకుఇవ్వు

Output format: As shown in Table 3.1 and the respective key is given in Table 1

Table 3.1: Telugu Parts-Of-Speech output

Telugu Parts-Of-Speech output						
Word	Lemma	POS tag	Suffix	Coarse POS	Gender	Type
నాకుఇవ్వు	నాకు	NN	ఇవ్వు	pn	any	pl

Table 3.2: Key for Telugu POS[1]

Tag type	Number of tags	Tags (refer Appendix Table-2 and Table-3)
Main POS Tag	25	JJ (Adjective), NN (Noun), VM (Verb-finite), . .
Coarse POS Tag	9	n (noun), num (number), unk (unknown). . .
Gender	6	any, f (female), m (male),null
Number	4	any, pl (plural), sg (singular), null
Person	5	1, 2, 3, any, null

3.9 Smart Filter

Smart filer plays a vital role in understanding each POS tagged label before proceeding to extract sentiment from the Telugu SentiWordNet. It is a method of filtering of the word list to omit a few words which are not useful in extracting the sentiment, for example, Proper Noun like city names, month names, etc. It also filters outs the redundant words (if any) which improves the performance of the system. This smart filter enhances the Telugu sentiment analysis in noise reduction in the text which are like typo errors or semantic errors. In this research, great attention was not given in text noise reduction by assuming the data parsed to be in semantically meaningful Telugu text or sentence.

3.10 Telugu SentiWordNet

The major focus in this dissertation is given in building a Telugu SentiWordNet using existing English SentiWordNet. Also, SentiWordNet of two Indian dialects (Bengali and Hindi) are used for the creation of Telugu SentiWordNet. Word polarity has proven trustworthy for these resources in the Ph.D. thesis of Amitava Das[57]. For stronger assessment of sentiment of each phrase and removal of ambiguities, various SentiWordNets techniques were used during resource development. These WordNets have synsets (set of synonyms) connected via a prevalent ID deprived of word-to-word immediate encoding. Every synset is given three statistical scores which specify Negative, Positive and Neutrality of the word. Two different methods are proposed to build the Telugu SentiWordNet they are, (i) Using Telugu Dictionary(ii) Use existing IndoWordNet.

3.10.1 Using English-Telugu Dictionary

For Telugu, there is no WordNet corpus openly accessible which forced to create a new WordNet for Telugu. Different Telugu dictionaries were used to extract the Telugu words, namely English – Telugu dictionary from Digital South Asia Library[4], University of Chicago, USA and English – Telugu from the University of Hyderabad[5], India. After collecting the words from these dictionaries, a merger function was used to extract only unique words. There were about 132453 Telugu words in the format as shown in Table 3.3

Table 3.3: Results of English-Telugu dictionary

English-Telugu Dictionary result					
Word	transliteration	Language Code	POS tag	English meaning(s)	Synonyms
కరణి	karaṇi	Tel	Noun.	Manner, way, mode	విధము, కరణము

The sentiment labels like positive and negative for the Telugu words are extracted

[4] https://dsalsrv04.uchicago.edu/dictionaries/brown/ accessed on 05-July-2019

[5] http://caltslab.uohyd.ac.in/English-Telugu-Dictionary.html accessed on 05-July-2019

from their English equivalents SentiWordNet Interface[6] built by NLTK group. A string equivalent method is used to map the English term in the English-Telugu dictionary and mapped with English SentiWordNet. The process is shown in steps and Figure 3.4:

1. Step-1: Collect English-Telugu words from the mentioned dictionaries.
2. Step-2: Filter the unique Telugu words from the data collected in step-1.
3. Step-3: Fetch English word in the obtained English-Telugu word.
4. Step-4: Loop each word collected from step-3 over the NLTK English SentiWordNet.
5. Step-5: If words matches take the polarity and sentiment label from NLTK English SentiWordNet.
6. Step-6: Store it in a local file.
7. Step-7: Repeat the process for the entire wordlist from step-2.

This approach resulted in 42,365 Telugu SentiWordNet records with multiple synset (set of synonyms), Parts of Speech tags, polarity, and sentiment labels.

3.10.2 Using IndoWordNets for Indian Languages

“Wordnets are lexical structures composed of synsets and semantic relations” [58]. It is a machine-readable English linguistic lexical library created by Princeton University. It has become the most useful tool for the implementation of natural language processing. A synset comprises a set of synonyms. Government of India and Indian Institute of Technology, Bombay (IITB) collaborated to create an IndoSentiWord which includes most of the Indo-Aryan languages. However, Telugu belongs to the Dravidian language group with a similar dialect of Hindi and Bengali. These WordNets are linked across languages through common synset IDs. Figure 3.5 show the overall implementation of this approach.

[6] http://www.nltk.org/howto/sentiwordnet.html accessed on 05-July-2019

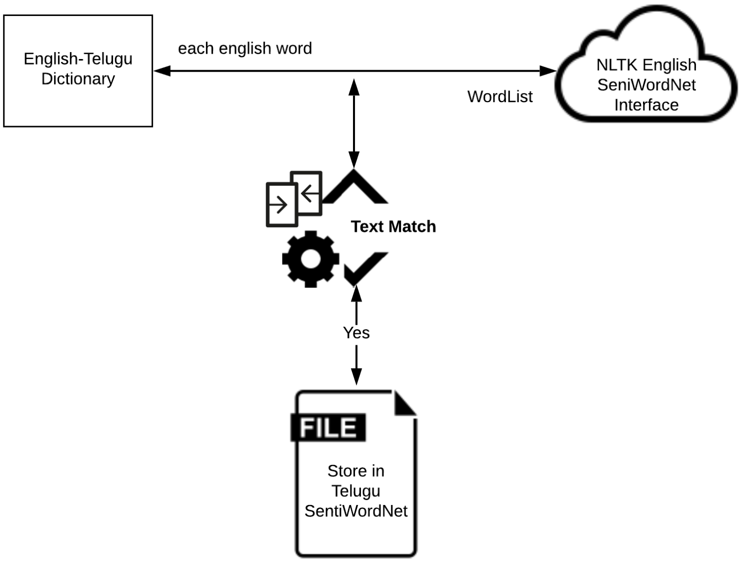

English-Telugu Dictionary
each english word
WordList
NLTK English SeniWordNet Interface
Text Match
Yes
FILE
Store in Telugu SentiWordNet

Figure 3.4: Overview of Telugu SentiWordNet using dictionary

1. Step-1: The development of Source Lexicon starts with collecting the SentiWordNets for the three Indian languages mentioned above. Words polarity are obtained from their respective SentiWordNets.

2. Step-2: From the above-collected WordNet, the corresponding synset ID of each word is extracted. This is achieved with the use of a hash-map for all the words in that language's WordNet.

3. Step-3: Put the each identified synset ID as δ 'key' in the hash-map and polarity as the value of the hash-map. In case if there is a key present for the identified synset ID, then append-only polarity to the hash-map.

4. Step-4: A certain key (Synset ID) is considered as the final polarity of its list of synset IDs with a majority greater than 65 percent. This leads to a list of synset IDs with a significant polarity attributed.

5. Step-5: Telugu WordNet is collected for generating the Target Lexicon, from the derived hash-map in step-3, loop over each Telugu Wordnet and find each

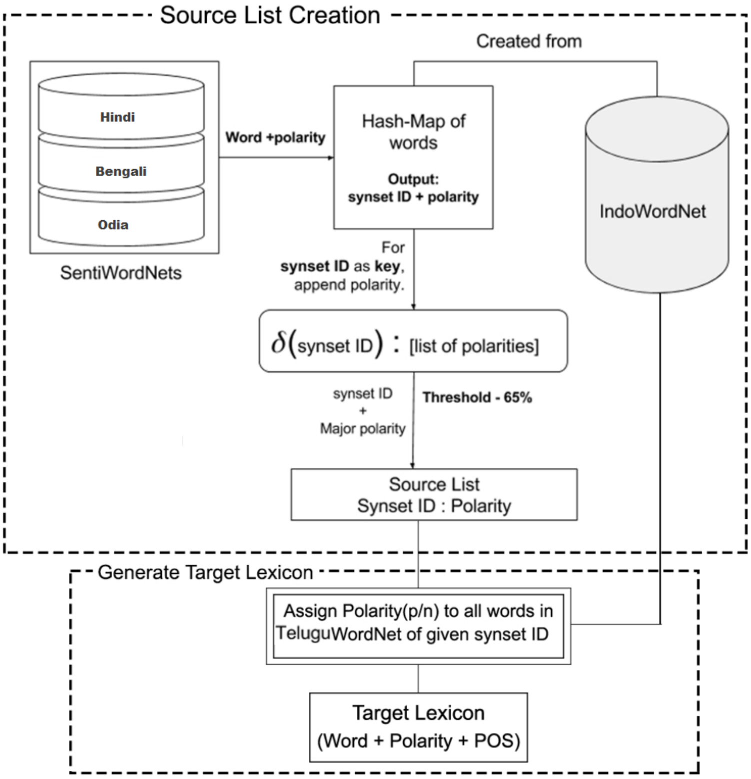

Source List Creation
Created from
Hindi
Bengali
Odia
SentiWordNets
Word +polarity
Hash-Map of words
Output: synset ID + polarity
IndoWordNet
For synset ID as key, append polarity.
δ(synset ID) : [list of polarities]
synset ID + Major polarity
Threshold - 65%
Source List Synset ID : Polarity
Generate Target Lexicon
Assign Polarity(p/n) to all words in TeluguWordNet of given synset ID
Target Lexicon (Word + Polarity + POS)

Figure 3.5: Overview of Telugu SentiWordNet using IndoWordNet

synset. If an equal synset is found extract the polarity and appended to the Telugu lexicon.

After execution of the above steps, the final Telugu lexicon will be comprised of words with sentiment polarity, pos tag, and synset ID.

3.10.3 Evaluation of the developed Telugu SentiWordNet

In order to evaluate the above-proposed method for Telugu SentiWordNet, randomly 1500 words were picked from the obtained lexicon. This includes positive, negative and neutral words for balancing the evaluation. Three annotators who are proficient in speaking, writing and reading the Telugu language were chosen to estimate the polarity of the words. No annotator had previous data on the polarity allocated to a token. Thus, makes the annotation of the tokens fair and balanced. Fleiss Kappa's ranking for the annotated sample collection was also calculated to record inter-annotator consent. The below equation is used to calculate Fleiss Kappa:

$$\kappa = \frac{\overline{P} - \overline{P}_e}{1 - \overline{P}_e} \tag{3.1}$$

Where $\overline{P}$ represents the sum of observed agreement. The sum of an agreement by chance is denoted by $\overline{P}_e$. The Telugu SentiWordNet recorded a significant agreement score of κ=0.74.

3.11 Analysis of the developed Telugu SentiWordNet

As the main objective of this dissertation is to perform sentiment analysis on the Telugu language, a lexicon of Telugu SentiWordNet is mandatory for achieving this goal. Hence, multiple strategies were considered and evaluated with consciousness and finally designed the above explained two methods which seemed to be perfect for building Telugu SentiWordNet. The first method focuses on English-Telugu dictionary pair in which each English meaning and their synset from the dictionary pair is matched with the English SentiWordNet and a new Telugu lexicon was generated. This process is reliable in terms of consistency between English SentiWordNet and derived Telugu SentWordNet with equal polarities. Though this is an easy process of attaining Telugu SentiWordNet, unfortunately, Telugu script is very different in writing and usage of words inappropriate sentences when compared with the English language. Also, Telugu is a Subject-Object-Verb (SOV) order whereas English is a Subject-Verb-Object (SVO) which make word usage in different cases and sentences. The valuation of this approach was simpler by just performing an equal's check of the word. In the first phase development, it was clearly understood that this would fail while executing Telugu nested sentences. Hence decided to go with the procedure of building SentiwordNet of Indo-Aryan language family.

Indo-Aryan language group like Hindi, Bengali, Odia, etc., have some similarities to a Dravidian language group, so analyzed the existing IndoWordNetfor these languages and made a decision to go with this approach of building a SentiWordNet for the Telugu language. The synset was mapped and extracted the polarities and a new Telugu lexicon was generated and have better results in execution than the other one. The only evaluation process for this method is to evaluate manually with the native speakers, this was tested by three annotators with Fleiss Kappa's inter-annotator agreement score of 0.74.

For the evaluation of the proposed design for Telugu sentiment analysis, the results from the second approach i.e Telugu SentiWordNet based on IndoWordNet was selected for implementation and the evaluation purposes. Also, this Telugu SentiWordNet is a helpful tool for any Telugu Data Sentiment Analysis. This technique is general and could be used to develop similar sentiment lexicons in the IndoWordNet framework for other Indian languages.

3.12 Judgement Function and Polarity extraction

Once the POS tagged sentences are parsed through the discussed Telugu SentWordNet (in section 3.10), the end result would have the word with POS tag, sentiment label, and polarity. Based on the obtained results a judgment would be made whether the given sentence is positive or negative. In this dissertation, for simplicity omitting neutral words or sentences as they hold less sentiment. There can be three cases to know the given sentence is positive or negative they are:

Case-1: All Positive words:

In this case, if all resultant words/sentence contains only positive words which are straight and easy to assume to be the given words/sentence to be positive. And polarity would be ‘1’

Case-2: All Negative word:

In this case, if all resultant words/sentence contains only negative words which are straight and easy to assume to be the given words/sentence to be negative. And polarity would be ‘-1’

Case-3: Mixed Sentiment

In this case, there would be a chance of having both positive and negative words in the given sentence. This case would be very frequent in the real-time scenario as most of the sentences in news articles or blog would discuss both the pros and cons of the topic. In that instance, the feeling of the phrase is difficult to decide. To handle this case, a counter function is added to keep a record on the word's sentiment labels. A hash-map is initialized, in which Sentiment label as key and word count as value, the value gets incremented for every time a word with same sentiment label is found. Calculate the sentiment score by the below equation:

$$PositiveRatio = \frac{Totalnumberofpositivewordsinthesentence}{Totalnumberofwordsinthesentence} \tag{3.2}$$

$$NegativeRatio = \frac{Totalnumberofnegativewordsinthesentence}{Totalnumberofwordsinthesentence} \tag{3.3}$$

The above equations 3.2 and 3.3 will calculate the individual sentiment of both positive and negative words which would be key in extracting the sentiment score. Sentiment score distinction can be obtained from Positive Ratio and Negative Ratio.

$$Sentiment_Score = PositiveRatio - NegativeRatio \tag{3.4}$$

The final sentiment score is calculated from the above equation 3.4, this score is vital in judging the sentiment of text. If the score is below '1' that means the given text is ***negative***, it the score is greater than '1' that the given text is ***positive*** or if the score is equal to '1' that means the text is ***neutral***.

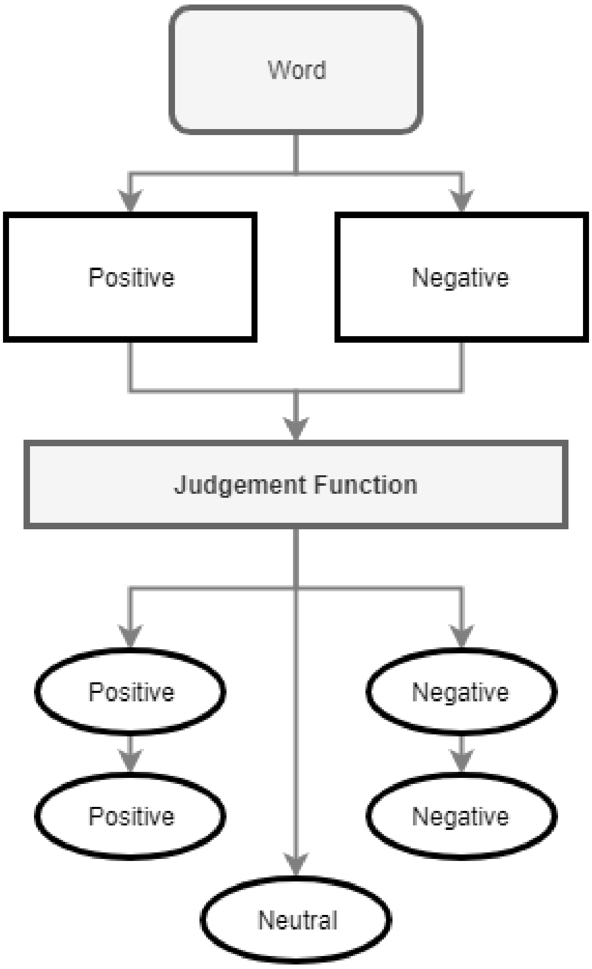


Figure 3.6: Judgment function flow of Telugu Sentiment extraction

3.13 Summary

This chapter gave an overview of design strategies made for building Telugu sentiment analyzer using lexicon-based methodology. Depicted the design diagram and explained each component for developing a sentiment analyzer. Discussed the adopted POS tagger [56] and its benefits and limitations. An overview of Telugu SentiWordNet construction using different approaches is discussed and given an analysis. Finally, the judgment function for calculating polarity score and identify sentiment label are discussed.

Chapter 4

Implementation

This chapter provides a clear understating of technical architecture and data-flow for the proposed system in the previous chapter-3. Wide range of technologies was adopted for building a web application. The designed web application provides a User Interface for taking input from the user and compute the text for applying text analysis and sentiment analysis. This makes the system unique in terms of sentiment analysis irrespective of domain or category. At the outset, data-flow is discussed on how the system components are organized with the selected technologies. Then an algorithmic flow is given for the better explanation of the system flow. This section is thus appropriately divided into three distinct sections, with each section providing a comprehensive overview of the implementation employed by that phase of the study. Also, gives an overview of the technical challenges encountered during the development phase.

"A good idea is about ten percent and implementation and hard work is 90 percent". -Guy Kawasaki

4.1 Functional and Data-flow

A functional proposal is a structured specification used for describing the expected capability, appearance, and relationships of a designed product. For implementing the designed system, initially many functional decisions were made for handling various cases, but unfortunately few design choices resulted in low accuracy. Finally, after a trial and error Figure4.1 was promising for achieving the goal of building a Senti-

WordNet based sentiment analysis for the Telugu language. This also describes both functional and data-flow in the system. This system has a two-step mechanism with five important components to extract sentiment from a given written Telugu script/text. The two-steps are:

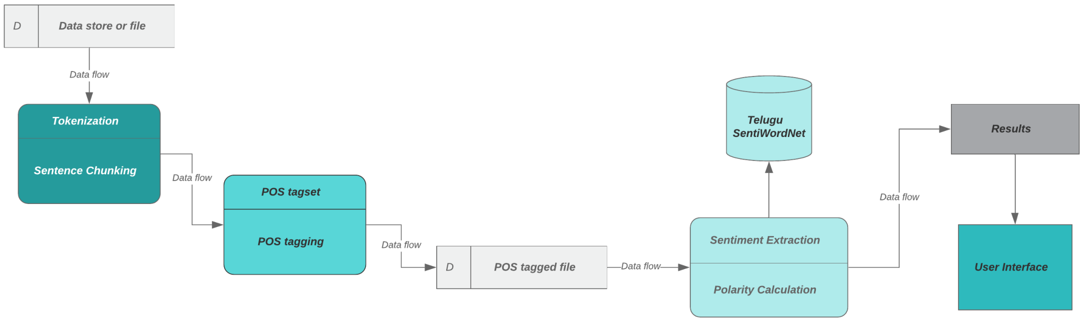


Figure 4.1: Data-Flow overview

- POS tagging
- Sentiment Extraction

The dataflow starts by taking an input Telugu text in the form of free text or a dataset, the data is pre-processed for removing illegal characters or words. The data input is set into a JSON and parsed to the server, firstly, JSON request is converted to a variable or object and an API call is made to Google's Cloud Natural Language API[1], the resulting output is expected to a language code, in this case, it would be **'Telugu: te'**. Then the data is tokenized and iterated over the POS tagger component in which a POS tag is identified for each word. The resultant data from POS tagger is iterated over the built Telugu SentiWordNet for extracting a sentiment label and polarity score of each word. Finally, polarity is calculated and results are shown in the user interface with a sentiment label and polarity score.

In order to evaluate the proposed idea/method a practical implementation and evaluation is mandatory. Figure 4.2 gives an overview of the technical implementation of the proposed method in the previous chapter-3. As shown, this architecture has three major components:

[1] https://cloud.google.com/translate/docs/reference/rest/v2/detect

- User Interface
- Server-side implementation
- POS tagger and Telugu SentiWordNet

4.2 Technical Implementation

This section gives an overview of technical design, chosen technologies, various implementation steps and the challenges that were faced during the development phase. This prototype is developed in Python programming language. Figure 4.2 depicts the technical architecture and technologies adopted for the development of the proposed system.

4.2.1 Python Programming Language

Python is a dynamic programming language of elevated levels, interpretation, and general-purpose, focusing on readability of the code. The word ”free software” with Python language relates to freedom to operate, edit, share, research, alter and enhance software. The syntax in Python enables software developers to code faster and fewer steps when compared with Java or C++. This programming language generally includes functional programming that is both imperative and objective. It has an extensive and powerful standard library with automated storage and realistic functionality. It impacted in much-developing software such as gaming, web apps, cognitive development and prototyping, the implementation for animations, etc., the Python language has diverse applications. This allows for a greater number of programming languages in the IT sector compared with other languages. Certain benefits of using python are [59][60][61][62]:

- It offers vast standard libraries covering fields such as string activities, the Internet, OS interfaces and protocols.
- Most of the extremely utilized programming functions have been scripted in it, limiting the length of Python code.

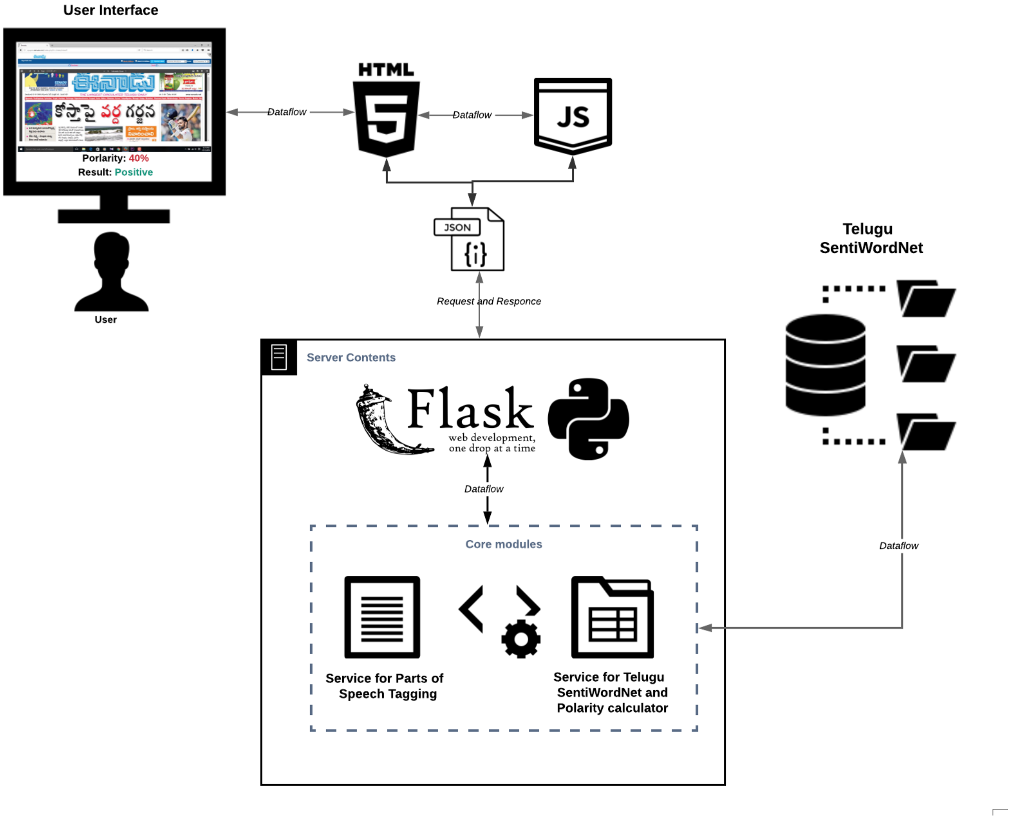


Figure 4.2: Technical Architecture

- The command capacity is strong when it calls C, C++ or Java via Jython effectively.
- Python also works with XML and other mark-up languages, because they can run with the same byte code for all advanced operating systems.
- In addition to providing a clean syntax, Python adds by grouping code into logical categories, like modules, classes, and features to create maintainable software.
- Although Python is often quick enough to meet many computation requirements, it still allows multidimensional loops to speed up the software.

This above advantages and ease-to-code syntax of python made to pick as the programming language for the development of this project. Python alongside Flask is adopted for web development of the prototype, which is a smooth combo of developing a web-based application with powerful libraries support. This acts as the server-side part of the system and processes the given input request and results out the sentiment of the given sentence with a polarity score.

4.2.2 Python Flask

Flask is a Web Gateway Interface (WSGI) toolkit and jinje2 model engine with a baseband Python-write microweb structure. It is categorized as a microframework because no specific tools or libraries are needed. It has no memory integration level, structure validation or other components with popular features from pre-existing third-party databases. Extensions can nevertheless be used to attach such features. These system facilities comprise an embedded HTTP server, device screening and the internet service RESTful [63]. The heart of this flask framework is ***server.py***, which acts as a controller between input and output requests.

Pre-requisites

- Python 2.7+,
- Flask: Python web application framework,
- Flask-CORS: A flask extension to handle cross-origin AJAX,
- UNIX GCC, Make
- JavaScript library
- Integrated Development Environment (IDE) with python extension in case of Visual Studio code.

Once, the above technologies were incorporated then configure the server endpoints in the ***server.py***, Flask by defaults supports HTTP RESTful framework. This framework makes easy to run a web application. Execute ***python server.py*** command in the python console to run the application, by default a static HTML routing is provided and it is hosted on the local system with default IP address and port number

127.0.0.1:5000 or localhost:5000. Listing 4.6 shows the code that would execute on running the ***python server.py*** command in the python console.

Listing 4.1: Python code for starting the server with default routing

```
@app.route('/')
def index():
    return render_template('teluguSentimentAnalyser.html')
```

On executing this code, it would land on ***teluguSentimentAnalyser.html*** which is a static web-page in User Interface (UI).

The UI design is often discussed in combination with user experience (UX), which may involve the stylistic presence of the computer, response time and user-friendly information. This component plays a very important role in taking input from the user and process it to sever and display the logical output from the processed data. In this system, a user interface is designed for ease of use from an end-user perception. Different technologies were considered for building the UI of this system for better user experience. The technologies adopted include HTML for building static webpages, JavaScript for adding dynamicity for the HTML pages, Angular JS and CSS for making the UI powerful by handling client-side validations and styles. The design has three important elements, (i) Input Fields/Controls, and (ii) Informational component. The use of each component is explained below:

- (i) Input Fields/Controls: This component contains an input box <input>, this box is used to take dynamic input from the user with customized CSS re-sizing property as shown. This CSS property makes the input box more user-friendly to view the entered text to any intent of the screen window. Also, includes buttons to control the data flow to the server.

- (ii) Informational component: This component displays the data in the HTML webpages, there are many different ways to visualize data on the UI. Aim of this dissertation is to know the sentiment of the given text/sentence in which polarity plays a vital role in the subjective analysis. The subjective analysis includes both positive and negative sentiment polarities, where polarity score is extracted from the text/sentence which is a numeric value. In order to show the user how much percentage is the given text/sentence is positive or negative, a percentage indicator bar as depicted in Figure 4.3 is displayed in the UI. Also, a resultant text which is a sentiment label for the requested text/sentence is shown.

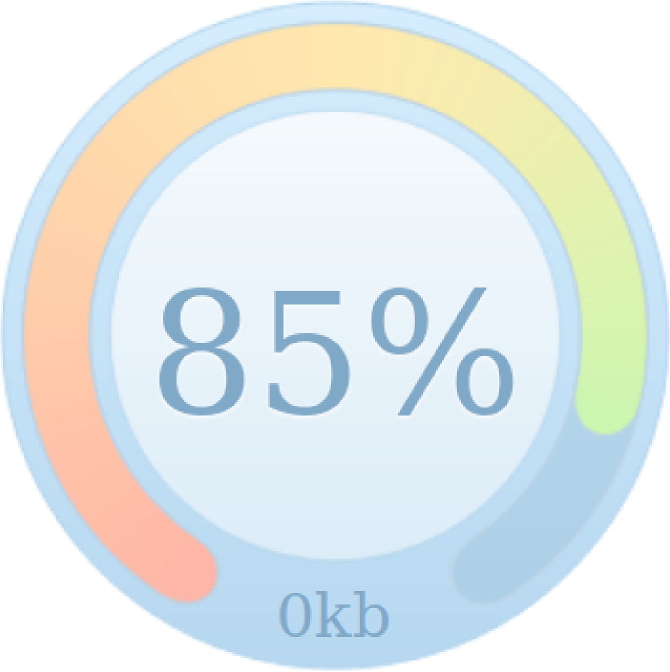


Figure 4.3: Polarity score indicator

Once, the Telugu data is entered into the input box, an Angular JS validation is done to verify the requested data doesn't contain any illegal arguments which are not expected for data processing. On click of the compute sentiment button, a json request is framed with the inputted text/sentence.

JSON request

var inputData = `` ఏపీ ముఖ్యమంత్రి జగన్ ప్రధాని మోదీతో సుదీర్ఘంగా భేటీ అయ్యారు. పార్లమెంట్లోని ప్రధాని కార్యాలయంలో జరిగిన ఈ సమావేశంలో ఎంపీలు..అధికారులతో సమీక్ష ముగిసిన అనంతరం ప్రధాని..ముఖ్యమంత్రి ఇద్దరూ దాదాపు అరగంట సేపు ఏకాంతంగా చర్చలు చేసారు."

Listing 4.2: JSON input

```
{"data": {
  "id": "X123",
  "value": inputData
  }}
```

Then a **jQuery.ajax()** POST call to ***“sentiment”*** service as shown in Listing 4.3 is made for the processing of data from client to back-end, in this instance it is *localhost:5000 or 127.0.0.1:5000*

Listing 4.3: Code for jQuery.ajax() call to “sentiment” service in JavaScript

```
$.ajax({
    url: "http://127.0.0.1:5000/sentiment",
    type: "POST",
    contentType: "application/json",
    data: JSON.stringify({'input': inputData}),
    success: function(data){
                sentiment = data['data'] * 100;
    }
});
```

The parsed JSON request is processed by the service called for executing the sentiment analysis. As discussed in chapter-3, firstly, a Google Natural Language Cloud API is called for validating the requested input with the expected language to be ‘Telugu:te’. If the response is success then the data is passed to Parts-of-Speech (POS) tagging service to retrieve and append POS tag to each word. The POS tagging module is adapted from [56][2]. This POS tagging system is considered as a bench-marking system in the field of NLP for Telugu language and developed in collaboration with Sketch Engine[3], a company dedicated for building language corpus, text analytics, dictionaries etc. Execution steps are given below:

[2] https://bitbucket.org/sivareddyg/telugu-part-of-speech-tagger/src/master/

[3] https://www.sketchengine.eu/

4.2.3 GNU Make

GNU is a computer operating system and software collection. **Make** is among the simplest and most used techniques for build automation. Since Feldman's initial proposition in 1979 [64], there have been several applications. It offers numerous other characteristics such as variables, functions and conditionality's that enable developers to make guidelines more flexible. It is achieved by using **Makefiles**, which are an easy way to arrange a complete set of software, it governs automatically what parts of a big program must be recompiled and hitches orders to recompile them. In other words a **makefile** functions according to the principle that files only need to be reconstructed if the dependency of the file is less than created /recreated [65]. **Makefile** aims to compile and connect a program and steps in order to execute the program. Lines of control must be marginalized by tab characters, space causes errors since **make** has a case-sensitive issue that causes common errors.

Listing 4.4: Makefile code with make commands

```
TAGGER=./bin/tnt -H -v0 models/telugu
LEMMATIZER=./bin/lemmatiser.py models/telugu.lemma
TAG2VERT=./bin/tag2vert.py
TOKENIZER=./bin/unitok.py -l telugu -n

tag:
        cat telugu.input.txt | $(TOKENIZER) |
            sed -e 's/^\.$$/.\n<\/s>\n<s>/g' |
            sed -e 's/^\?$$/?\n<\/s>\n<s>/g' > telugu.tmp.words
        $(TAGGER) telugu.tmp.words | sed -e 's/\t\+/\t/g' |
            $(LEMMATIZER) | $(TAG2VERT) > telugu.output.txt
        rm telugu.tmp.words
        echo "Output␣stored␣in␣telugu.output.txt"
```

Step 1: Read the input JSON data.

Step 2: Copy the "value object's data to a text file.

Step 3: Run command ***make tag*** which contains the execution steps internally.

Step 4: Output file with POS tags appended to each word will be generated.

Listing 4.5: Python code for executing POS tagging

```
#Opens the input text file
 with codecs.open('/telugu.input.txt','rwa+', encoding = 'utf-8') as inputfile:
    inputfile.truncate()
    for i in sentence:
        inputfile.write(i.decode('utf-8'))
        \\Writes to input text file
        inputfile.write("␣")
inputfile.close()

#execute Make
os.system("cd␣telugu-part-of-speech-tagger;␣make␣tag")

#Opens the output text file
with open('/telugu.output.txt','r') as outputFile:
    #Read output file with POS tags
    output = outputFile.read()
```

The Sentiment extraction of this prototype works on the custom-built Telugu SentiWordNet with the standards of IndoWordNet and SentiWordNet. In total there were 42,351 Telugu SentiWordNet are annotated with different approaches as explained in chapter-3, the collected SentiWordNet has three types of sentiment labels and parts-of-speech tags associated with each word in the Telugu WordNet. Table 4.1 gives a distribution of the SentiWords and Figure 4.4 shows the Telugu-English combinational WordNet and derived Telugu SentiWordNets. Firstly, the parsed POS tagged sentence would be tokenized and each word would filtered out with the elimination of objective words as these words are constant and universal which doesn't contains any sentiment or emotion. This filtered list be looped over the Telugu SentiWordNet and extracts the polarity of the parsed data.

Step 1: Input POS tagged sentence.

Step 2: Tokenization

Step 3: Word categorization based on POS tag, example: Noun, Adjective

Step 4: Loop over SentiWordNet

Step 5: Judgement Function for Polarity extraction

Step 6: Total sentiment calculated from the equations (1), (2) and (3).

Step 7: Append Sentiment label and polarity of the sentence

Step 8: Return results

Table 4.1: Telugu SentiWord distribution

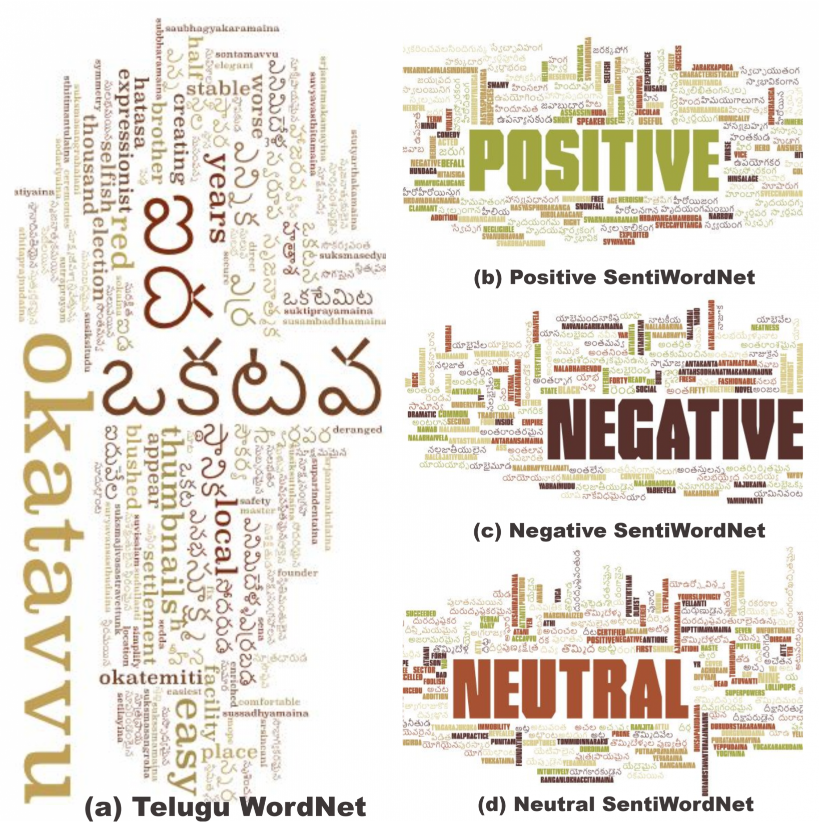
Telugu SentiWord collection				
POS type	**Negative**	**Positive**	**Neutral**	**Total**
Noun	2783	2164	3523	8470
Verb	1289	2049	3451	6789
Adverb	1616	2196	7256	11128
Adjective	2546	4313	5342	12201
Unknown/Others	1371	1198	1254	3828
Total	**9035**	**11920**	**20826**	**42351**

4.3 Analysis of the algorithm (Computational complexity)

Computational complexity is a computer science domain that analyzes algorithms depending on the number of resources necessary to run them. This measured in terms of space and time complexity.

4.3.1 Time complexity

The time complexity in computer science is the computational complexity which identifies how much time an algorithm takes for performance estimation. The number of primary operations performed by the algorithm is generally estimated by counting the time complexities, assuming that each fundamental operation takes a configured amount of time to execute.



(a) Telugu WordNet

(b) Positive SentiWordNet

(c) Negative SentiWordNet

(d) Neutral SentiWordNet

Figure 4.4: (a) Telugu-English Combinational WordNet and (b),(c) and (d) are Processed SentiWordNet

In general, the time complexity is calculated in two cases, they are: best case and worst case.

- Best case: This is the complexity of solving the problem for the best input.
- Worst case: This is the complexity of solving the problem for the worst input of size n

In general, the time complexity of an algorithm is estimated in terms of Big-O notation

4.3.2 Big-O Notation

Big-O notation, sometimes called “asymptotic notation”, is a mathematical notation that describes the limiting behavior of a function when the argument tends towards a

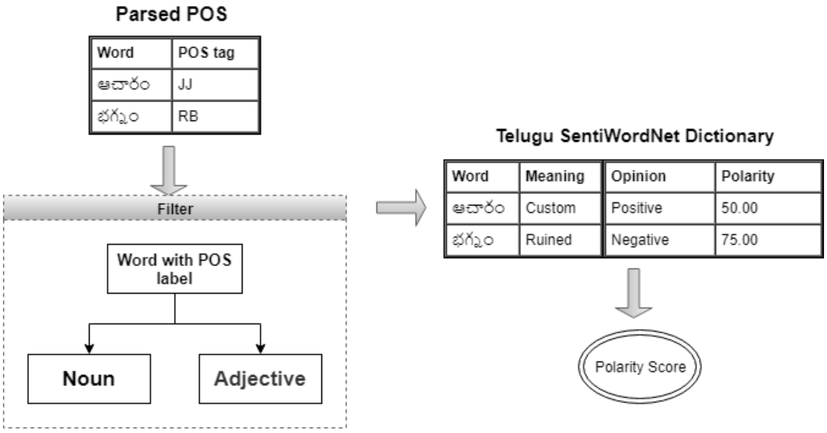


Figure 4.5: Overall dataflow with Telugu SentiWordNet

particular value or infinity. It is represented as ***‘O’***

The time complexity of the designed algorithm is evaluated in the below steps:

Assumptions:

- The adopted POS tagger is sorted and stored in a map using a sorting algorithm.
- The derived Telugu SentiWordNet has two files, Adjectives, and Adverbs. Assuming these two files are read and sorted into a map in variables ***adjectiveSentiMap, adverbSentiMap***.

Listing 4.6: Tokenization

```
inputSentence =‘‘Welcome to the Trinity”
tokens␣=␣inputSentence.split()
␣
```

- Firstly, the inputText is tokenized using python split function on space to get each word as a token

  Time complexity: The best case and the worst case time complicity for splitting a string is O(N), where N is the size of the sentence.

- Retrieve POS tag for each word from [56] module which is executed using a ***gcc make command***

Time complexity: Assuming that the dictionary data is stored in a map format. So the retrieval time complexity from a map is O(1) in both best and worst-case (as the retrieval from a map works based on the hashing algorithm).

- The resultant tokens with POS tags will be filtered for objectivity ans=d subjectivity classification in which proper nouns, determiners are removed from the map.

  Time complexity: O(1) in both best-case and worst-case scenario.

- Iterate POS tagged tokens over their respective ***adjectiveSentiMap, adverbSentiMap*** to extract sentiment and polarity of each POS tagged word. for key in tokenMap: d[key]=[POS tag, adjectiveSentiMap.get(key)] or d[key]=[POS tag, adverbentiMap.get(key)]

  Time complexity: Here again, fetch the details for each taken from a map of POS tagged tokens. To get the details for each token from adjectiveSentiMap or adverbSentiMap, it will be taking the O(1) time in both best and worst case.

So, the total time complexity that will be taken by this algorithm to get the polarity for each object is: Best case:

- O(N) – For splitting the sentence.
- O(1) – For identifying the POS tag of each token from a map or dictionary.
- O(1) – For iterating over adjectiveSentiMap or adverbSentiMap
- O(1) – For retrieving polarity details of those tokens from the adjectiveSentiMap or adverbSentiMap.

The final best-case time complexity will be: **O(N) + O(1) + O(1) + O(1)= O(N)**

Worst case:

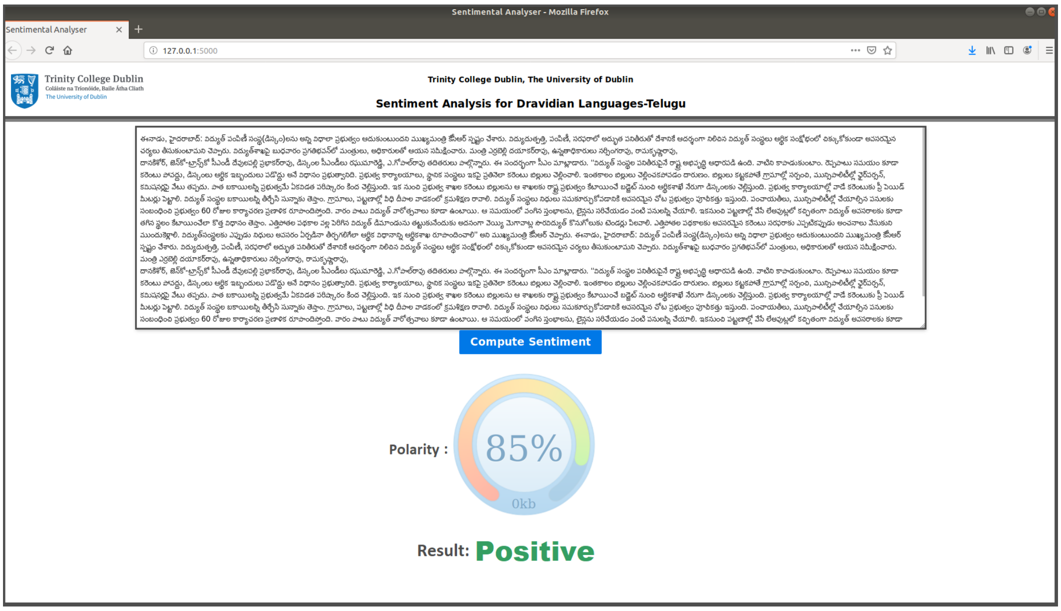
- O(N) – For splitting the sentence.
- O(1) – For identifying the POS tag of each token from a map of dictionary
- O(N) – For iterating over adjectiveSentiMap or adverbSentiMap

- O(1) – For retrieving polarity details of those tokens from the adjectiveSentiMap or adverbSentiMap.

The final worst-case time complexity will be: $\mathbf{O(N) + O(1) + O(N) + O(1) = O(N)}$

The time complexity for the above approach will be O(N) in both the best and worst case.

Finally, the resulted data consists of a subjective Sentiment label and polarity score in an python object. This data is parsed as JSON and sends as a response to the UI using Python Flask web framework as shown in Figure 4.6.



Trinity College Dublin, The University of Dublin

Sentiment Analysis for Dravidian Languages-Telugu

ఈనాడు, హైదరాబాద్: విద్యుత్ పంపిణీ సంస్థ(డిస్కం)లను అన్ని విధాలా ప్రభుత్వం ఆదుకుంటుందని ముఖ్యమంత్రి కేసీఆర్ స్పష్టం చేశారు. విద్యుదుత్పత్తి, పంపిణీ, సరఫరాలో అద్భుత పనితీరుతో దేశానికే ఆదర్శంగా నిలిచిన విద్యుత్ సంస్థలు ఆర్థిక సంక్షోభంలో చిక్కుకోకుండా అవసరమైన
చర్యలు తీసుకుంటామని చెప్పారు. విద్యుత్‌శాఖపై బుధవారం ప్రగతిభవన్‌లో మంత్రులు, అధికారులతో ఆయన సమీక్షించారు. మంత్రి ఎర్రబెల్లి దయాకర్‌రావు, ఉన్నతాధికారులు నర్సింగరావు, రామకృష్ణారావు,
దానకిశోర్, జెన్‌కో-ట్రాన్స్‌కో సీఎండీ దేవులపల్లి ప్రభాకర్‌రావు, డిస్కంల సీఎండీలు రఘుమారెడ్డి, ఎ.గోపాల్‌రావు తదితరులు పాల్గొన్నారు. ఈ సందర్భంగా సీఎం మాట్లాడారు. “విద్యుత్ సంస్థల పనితీరుపైనే రాష్ట్ర అభివృద్ధి ఆధారపడి ఉంది. వాటిని కాపాడుకుంటాం. రెప్పపాటు సమయం కూడా
కరెంటు పోవద్దు, డిస్కంలు ఆర్థిక ఇబ్బందులు పడొద్దు అనే విధానం ప్రభుత్వానిది. ప్రభుత్వ కార్యాలయాలు, స్థానిక సంస్థలు ఇకపై ప్రతినెలా కరెంటు బిల్లులు చెల్లించాలి. ఇంతకాలం బిల్లులు చెల్లించకపోవడం దారుణం. బిల్లులు కట్టకపోతే గ్రామాల్లో సర్పంచి, మున్సిపాలిటీల్లో ఛైర్‌పర్సన్,
కమిషనర్లపై వేటు తప్పదు. పాత బకాయిలన్నీ ప్రభుత్వమే ఏకవిడత పరిష్కారం కింద చెల్లిస్తుంది. ఇక నుంచి ప్రభుత్వ శాఖల కరెంటు బిల్లులను ఆ శాఖలకు రాష్ట్ర ప్రభుత్వం కేటాయించే బడ్జెట్ నుంచి ఆర్థికశాఖే నేరుగా డిస్కంలకు చెల్లిస్తుంది. ప్రభుత్వ కార్యాలయాల్లో వాడే కరెంటుకు ప్రీ పెయిడ్
మీటర్లు పెట్టాలి. విద్యుత్ సంస్థల బకాయిలన్నీ తీర్చేసి సున్నాకు తెస్తాం. గ్రామాలు, పట్టణాల్లో వీధి దీపాల వాడకంలో క్రమశిక్షణ రావాలి. విద్యుత్ సంస్థలు నిధులు సమకూర్చుకోవడానికి అవసరమైన చోట ప్రభుత్వం పూచీకత్తు ఇస్తుంది. పంచాయతీలు, మున్సిపాలిటీల్లో చేయాల్సిన పనులకు
సంబంధించి ప్రభుత్వం 60 రోజుల కార్యాచరణ ప్రణాళిక రూపొందిస్తోంది. వారం పాటు విద్యుత్ వారోత్సవాలు కూడా ఉంటాయి. ఆ సమయంలో వంగిన స్తంభాలను, లైన్లను సరిచేయడం వంటి పనులన్నీ చేయాలి. ఇకనుంచి పట్టణాల్లో వేసే లేఅవుట్లలో కచ్చితంగా విద్యుత్ అవసరాలకు కూడా
తగిన స్థలం కేటాయించేలా కొత్త విధానం తెస్తాం. ఎత్తిపోతల పథకాల వల్ల పెరిగిన విద్యుత్ డిమాండును తట్టుకునేందుకు అదనంగా వెయ్యి మెగావాట్ల సౌరవిద్యుత్ కొనుగోలుకు టెండర్లు పిలవాలి. ఎత్తిపోతల పథకాలకు అవసరమైన కరెంటు సరఫరాకు ఎప్పటికప్పుడు అంచనాలు వేసుకుని
ముందుకెళ్లాలి. విద్యుత్‌సంస్థలకు ఎప్పుడు నిధులు అవసరం ఏర్పడినా తీర్చగలిగేలా ఆర్థిక విధానాన్ని ఆర్థికశాఖ రూపొందించాలి” అని ముఖ్యమంత్రి కేసీఆర్ చెప్పారు. ఈనాడు, హైదరాబాద్: విద్యుత్ పంపిణీ సంస్థ(డిస్కం)లను అన్ని విధాలా ప్రభుత్వం ఆదుకుంటుందని ముఖ్యమంత్రి కేసీఆర్
స్పష్టం చేశారు. విద్యుదుత్పత్తి, పంపిణీ, సరఫరాలో అద్భుత పనితీరుతో దేశానికే ఆదర్శంగా నిలిచిన విద్యుత్ సంస్థలు ఆర్థిక సంక్షోభంలో చిక్కుకోకుండా అవసరమైన చర్యలు తీసుకుంటామని చెప్పారు. విద్యుత్‌శాఖపై బుధవారం ప్రగతిభవన్‌లో మంత్రులు, అధికారులతో ఆయన సమీక్షించారు.
మంత్రి ఎర్రబెల్లి దయాకర్‌రావు, ఉన్నతాధికారులు నర్సింగరావు, రామకృష్ణారావు,
దానకిశోర్, జెన్‌కో-ట్రాన్స్‌కో సీఎండీ దేవులపల్లి ప్రభాకర్‌రావు, డిస్కంల సీఎండీలు రఘుమారెడ్డి, ఎ.గోపాల్‌రావు తదితరులు పాల్గొన్నారు. ఈ సందర్భంగా సీఎం మాట్లాడారు. “విద్యుత్ సంస్థల పనితీరుపైనే రాష్ట్ర అభివృద్ధి ఆధారపడి ఉంది. వాటిని కాపాడుకుంటాం. రెప్పపాటు సమయం కూడా
కరెంటు పోవద్దు, డిస్కంలు ఆర్థిక ఇబ్బందులు పడొద్దు అనే విధానం ప్రభుత్వానిది. ప్రభుత్వ కార్యాలయాలు, స్థానిక సంస్థలు ఇకపై ప్రతినెలా కరెంటు బిల్లులు చెల్లించాలి. ఇంతకాలం బిల్లులు చెల్లించకపోవడం దారుణం. బిల్లులు కట్టకపోతే గ్రామాల్లో సర్పంచి, మున్సిపాలిటీల్లో ఛైర్‌పర్సన్,
కమిషనర్లపై వేటు తప్పదు. పాత బకాయిలన్నీ ప్రభుత్వమే ఏకవిడత పరిష్కారం కింద చెల్లిస్తుంది. ఇక నుంచి ప్రభుత్వ శాఖల కరెంటు బిల్లులను ఆ శాఖలకు రాష్ట్ర ప్రభుత్వం కేటాయించే బడ్జెట్ నుంచి ఆర్థికశాఖే నేరుగా డిస్కంలకు చెల్లిస్తుంది. ప్రభుత్వ కార్యాలయాల్లో వాడే కరెంటుకు ప్రీ పెయిడ్
మీటర్లు పెట్టాలి. విద్యుత్ సంస్థల బకాయిలన్నీ తీర్చేసి సున్నాకు తెస్తాం. గ్రామాలు, పట్టణాల్లో వీధి దీపాల వాడకంలో క్రమశిక్షణ రావాలి. విద్యుత్ సంస్థలు నిధులు సమకూర్చుకోవడానికి అవసరమైన చోట ప్రభుత్వం పూచీకత్తు ఇస్తుంది. పంచాయతీలు, మున్సిపాలిటీల్లో చేయాల్సిన పనులకు
సంబంధించి ప్రభుత్వం 60 రోజుల కార్యాచరణ ప్రణాళిక రూపొందిస్తోంది. వారం పాటు విద్యుత్ వారోత్సవాలు కూడా ఉంటాయి. ఆ సమయంలో వంగిన స్తంభాలను, లైన్లను సరిచేయడం వంటి పనులన్నీ చేయాలి. ఇకనుంచి పట్టణాల్లో వేసే లేఅవుట్లలో కచ్చితంగా విద్యుత్ అవసరాలకు కూడా

Result: Positive

Figure 4.6: Developed Telugu Sentiment Analyser web application UI

Chapter 5

Experiment and Results

This chapter is intended to perform a hands-on experiment of the developed system. It starts with data collection criteria, collecting e-newspaper articles and manual annotation for determining ground truth results. Execution of the acquired dataset on the developed system and find the accuracy. Also, discusses time performance, case study and several observations.

5.1 Data Curation

Criteria: A rigorous query criteria for news information was enforced, using only electronic newspapers from Telugu publications would be selected and must contain the terms KCR[1] and Politics, or Telangana Chief Minister and Elections(s) within the headline or opening paragraph.

5.1.1 Media concentration

There are almost 28 titles that are published in Telugu speaking states of Telangana and Andhra Pradesh including 16 published in other southern states of India with a total circulation of 20 million copies. The major e-newspapers are Namasthe Telangana[2],

[1] KCR-The chiefminister of Telangana: https://en.wikipedia.org/wiki/$K._{C}handrashekar_{R}ao$ (visited on 03-07-2019)

[2] Namasthe Telangana e-paper, https://epaper.ntnews.com/ (accessed on 28th-June-2019)

Nava Telangana[3], Eenadu[4], Sakshi[5], Andhra Bhoomi[6], etc.

5.1.2 Criteria for Data collection

Any developed system needs to be tested for knowing whether the requirements were met or how efficiently the system is working. Decisions for data collection are important in terms of system proficiency. Following some principles can yield in collecting efficient data, in this research a certain goals were set in data collection, they are:

- Source: Only trusted sites that are exposed publicly free on the internet.
- Data-type: Continuous or discrete data.
- Timeliness: A specific duration.

5.1.3 Telugu news data collection

Telugu news data is collected from popular newspaper publishers which are circulated in Telugu language script in the native states of Telangana and Andhra Pradesh by crawling the web. Data is collected from 1st June 2019 to 30th June 2019, in total 1050 sentences from different electronic newspaper publishers as mentioned in Table 5.1. This is the general data irrespective of any domain or category.

5.2 Annotation

The dataset collected is given is to three native Telugu speakers who are expert in Telugu language with the highest degree in their respective fields. The annotators are Professors for Telugu language and belongs to Telangana and Andhra Pradesh states. They were given a task to annotate manually with positive, negative and neutral sentiment to each sentence in the dataset at their knowledge as shown in Table 5.2. The assessment by various annotators on the sentiment data in the dataset is extremely

[3] Nava Telangana e-paper, http://epaper.navatelangana.com/ (accessed on 28th-June-2019)

[4] Eenadu e-paper, http://epaper.eenadu.net (accessed on 28th-June-2019)

[5] Sakshi e-paper, https://epaper.sakshi.com/ (accessed on 28th-June-2019)

[6] Andhra Bhommi e-paper, http://epaper.andhrabhoomi.net/ (accessed on 28th-June-2019)

Table 5.1: Sources of Telugu e-Newspapers for data collection

Telugu Newspaper Corpus	
Name of the e-Newspaper publisher	**Total number of sentences**
Namaste Telangana	192
Mana Telangana	186
Velugu	153
Eenadu	187
Andhra Prabha	171
Manam	161
Total	**1050**

subjective. Though the annotators are experts in Telugu, there can be a chance of disagreement between annotators. So, it was decided to use metrics to quantify these agreements, to figure out how much annotators accept in assigning a specific note[66]. Cohen's kappa [67] is Popularly used in the classification of products into recognized, mutually exclusive classifications, the level of agreement between annotations. With the coefficient of Cohen Kappa, achieved the inter-annotative agreement with an annotative consistences (k-value) of 0.88. The annotated data is used in evaluating the final results by setting this as ground-truth results.

Table 5.2: Manual sentiment estimation by annotators

Ground-truth Results				
Name of the e-Newspaper publisher	**Negative**	**Positive**	**Neutral**	**Total**
Namaste Telangana	94	35	63	192
Mana Telangana	56	42	88	186
Velugu	48	26	79	153
Eenadu	90	54	43	187
Andhra Prabha	68	28	75	171
Manam	78	13	70	161
Total	**434**	**198**	**418**	**1050**

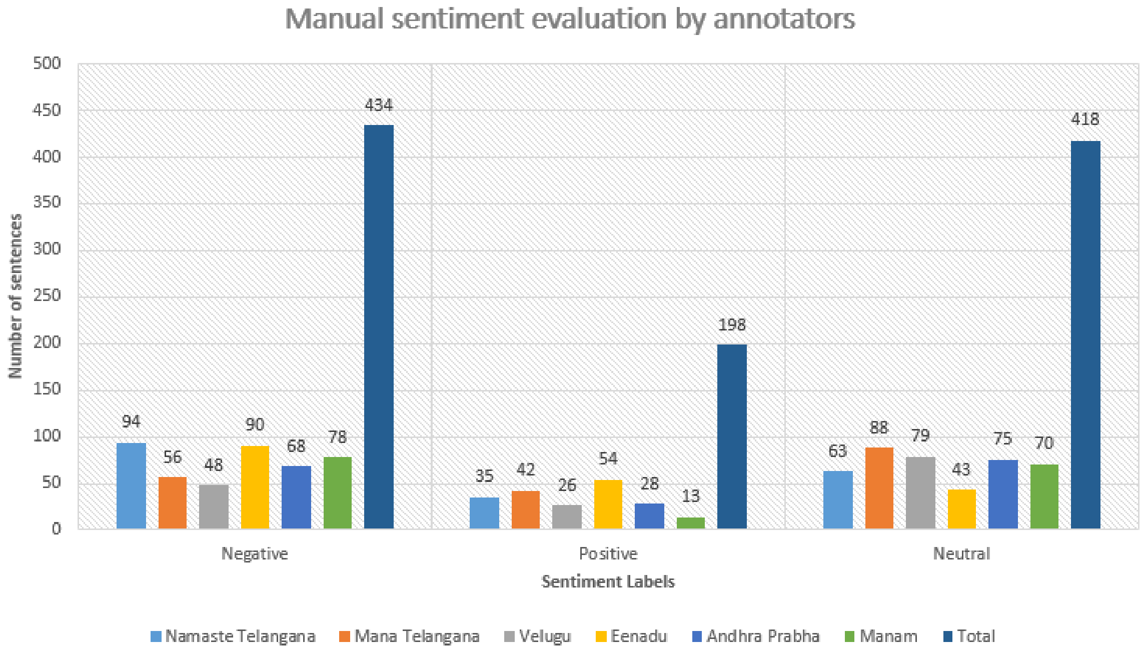


Figure 5.1: Manual sentiment estimation by annotator

5.3 Evaluation

The dataset framed is fed on the proposed method as discussed in chapter 3, initially, each sentence from the dataset is looped over a Trigrams-n-Tags (TnT) tagger for appending its respective parts-of-speech tag. Once, POS is tagged each word of the sentence is validated with Telugu SetiWordNet, in this process, it classifies each word as positive or negative or neutral. In this process, the retrieved word data is kept in a dictionary with the sentiment tag as key and a object of its count and polarity. This helps in performance improvement if there is a redundant word occurs in the sentences. Also, it would be easy to calculate the total polarity of the sentence. In general, the neutral words are objective in nature which doesn't contain much sentiment, for example, proper nouns, here all the neutral words possess a minimum polarity. The overall flow is shown in the Figure 5.2

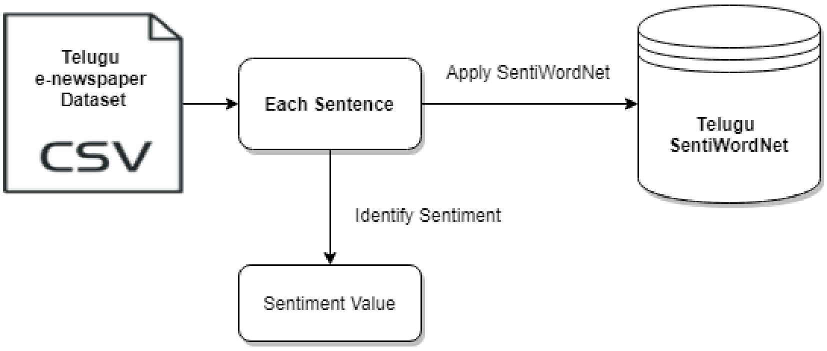


Figure 5.2: System execution flow

Table 5.3: Results of sentiment evaluation by the developed system

System Results				
Name of the e-Newspaper publisher	**Negative**	**Positive**	**Neutral**	**Total**
Namaste Telangana	82	22	88	192
Mana Telangana	47	38	101	186
Velugu	42	19	92	153
Eenadu	84	49	54	187
Andhra Prabha	58	22	91	171
Manam	58	07	96	161
Total	**371**	**157**	**255**	**1050**

5.3.1 Confusion Matrix

To evaluate the developed prototype, it was decided to draw a confusion matrix for the obtained results. A confusion matrix draws a table of classification which is often employed to define the results of a scoring system on a collection of test data assumed to be the actual scores[68].

Important terms to be noted in confusion matrix are:

- Positive (P) : Observation is positive.
- Negative (N) : Observation is not positive.
- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.

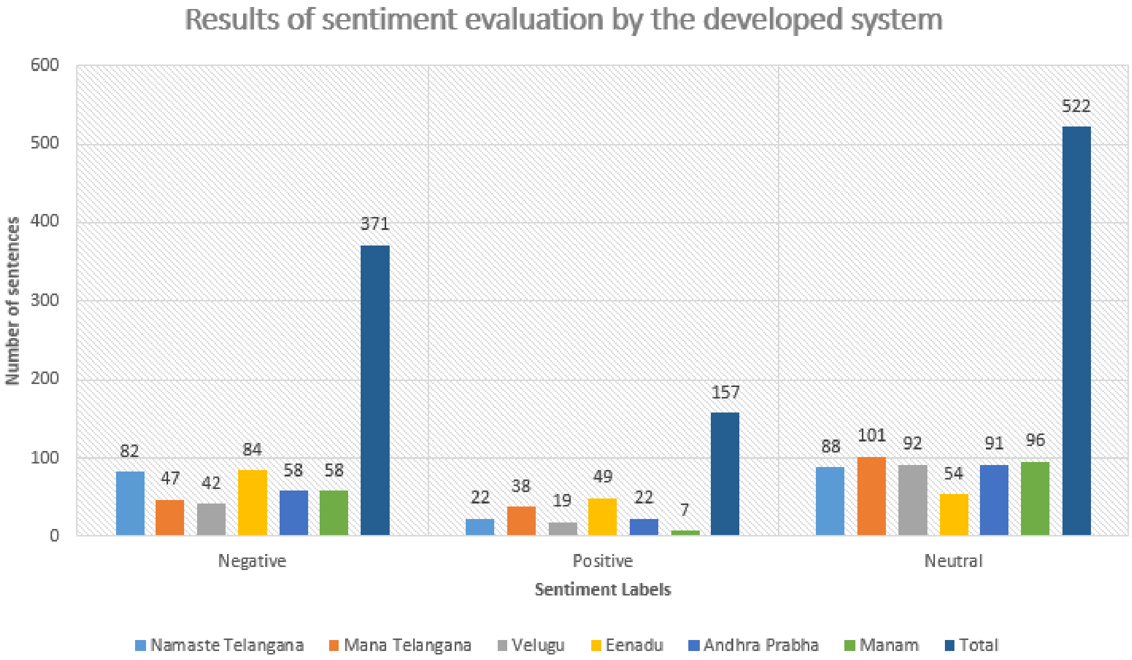


Figure 5.3: Results of sentiment evaluation by the developed system

- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.

The evaluation of a confusion matrix is computed in terms of ***accuracy***, ***precision***, ***recall***, and ***F1 score*** for a binary classifier [69]:

Accuracy –accuracy is the most appealing metric of results and is merely an equivalent of properly expected error to the overall findings in the **??** matrix. You might believe that our system is better if we have elevated precision. Yes, precision is a tremendous metric but only when symmetrical data sets exist, with attributes that are almost identical for false positives and false negatives. For this reason, other parameters should always be examined to assess the model's efficiency.

$$Accuracy(\%) = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})} \tag{5.1}$$

Precision: This shows the percentage of the positive elements identified that are right. Precision reports efficiently the overall subsequent likelihood and is thus a significant indicator of efficiency for unusual occurrences and formulated as shown in equation 5.2. Precision incorporates positives as well as negative results of samples and thus is a class prior dependent. It is often called purity.

$$Precision = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})} \tag{5.2}$$

Recall:Recall is the proportion of accurately predicted observations to all real category findings, as shown in the Table 5.3.

$$Recall = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})} \tag{5.3}$$

F1 Score is the median weighted of the precision and recall. This rating therefore requires into consideration both false positives and false negative It's not as simple to comprehend intuitively as precision, but F1 generally is better than precision, particularly if the category allocation is irregular. Precision operates better if false positive and false negative costs are comparable. If the costs are very distinct for false positives and false negative ones, both Precision and Recall are easier to be taken in the Table **??**.

$$\mathrm{F1} - Score = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{5.4}$$

5.4 Observations

On processing of the resultant data from the developed system and the ground-truth acquired from the manual annotation by Telugu language experts the below obervations were made:

- Total subjective phrases of the **632** are regarded to be positive and negative. **198** are positive and **434** are negative.
- The system formulated recognized **157** sentences as positive that is True Positive (TP), in this case ground truth was **198**.

- The ground truth was 434. It recognized the 371 phrases as negative (True Negative (TN).
- **41** is the False Negative FN that is negative but positive and
- **63** is False Positive FP, that is positive but negative

A confusion matrix is drawn from the above observations for statistical classification of the Telugu sentiment analyzer as given in Table 5.4.

Table 5.4: Confusion Matrix

Observations			
True positive (TP)	**True Negative (TN)**	**False Positive (FP)**	**False Negative (FN)**
157	371	41	63

By applying the above metrics for calculating ***accuracy***, ***precision***, ***recall***, and ***F1 score*** using equations 5.1, 5.2, 5.3, 5.4 has resulted in the subjectivity classification of the developed Telugu Sentiment Analyzer. Table 5.5 gives the calculated results:

Table 5.5: Subjectivity Analysis Results

Accuracy (%)	**Precision**	**Recall**	**F1-Score**
83.50%,	0.71	0.79	0.74

5.5 Performance Evaluation

5.5.1 Time performance

Performance analysis and evaluation will provide grounds for program execution. Knowing the total execution timespan shows how much improvement it requires to be. Understanding the execution duration of different program components enables to define the correct places for program modifications. Asymptotic performance testing is often a useful effectiveness guide. The important to efficiency is generally to do fewer operations. But brute force can improve the application in a few instances. In order to estimate the time performance of the developed Telugu sentiment analyzer is key in knowing how well the written algorithm works and shows that need to be improved.

To calculate the time performance of the developed system, a small experiment is conducted and executed on the prototype.

5.5.2 Prerequisite

Below configurations were used for analyzing the performance of the Telugu Sentiment Analyzer:

- CPU: Intel® Core™ i7-7700K CPU @ 4.20GHz
- Operating System: Ubuntu 18.06
- RAM: 16.00 GB

5.5.3 Experiment setup and assumptions

- Collected 5 samples each of different word length in the text i.e. 100, 500, 1000, 5000-word length Telugu sentences. The five sets of the collected data have different words and overall a different sentence but length is constant.
- Executed the experiment with each sample for 5 times and noted the recordings.
- Find the mean of the recorded readings by equation 5.5

$$Mean = \frac{SumofAllDatapoints}{TotalNumberofDatapoints} \tag{5.5}$$

- As previously discussed in chapter-3, POS tagger for this system is adopted from [56]. Hence, the performance calculation for POS tagging is calculated separately.
- This dissertation is focused on Telugu SentiWordNet, special attention is given to calculate the time performance of this system.

5.5.4 Execution

Parts-Of-Speech tagger by [56] is integrated into this system and an individual time evaluation is done for analyzing the performance of the system. Table 5.6 and Figure 5.4 shows the results of the POS tagger, where ***n*** is the nth sample.

Table 5.6: Average time taken for POS tagger execution

POS tagger observations						
Number of words in a sentence	**n=1**	**n=2**	**n=3**	**n=4**	**n=5**	**Average time**
100	3.4	2.9	4.3	3.8	4.1	3.7
500	5.3	4.6	6.02	5.89	7.52	5.866
1000	8.6	7.3	9.56	8.02	9.49	8.594
5000	11.34	10.4	12.03	13.97	12.54	12.056

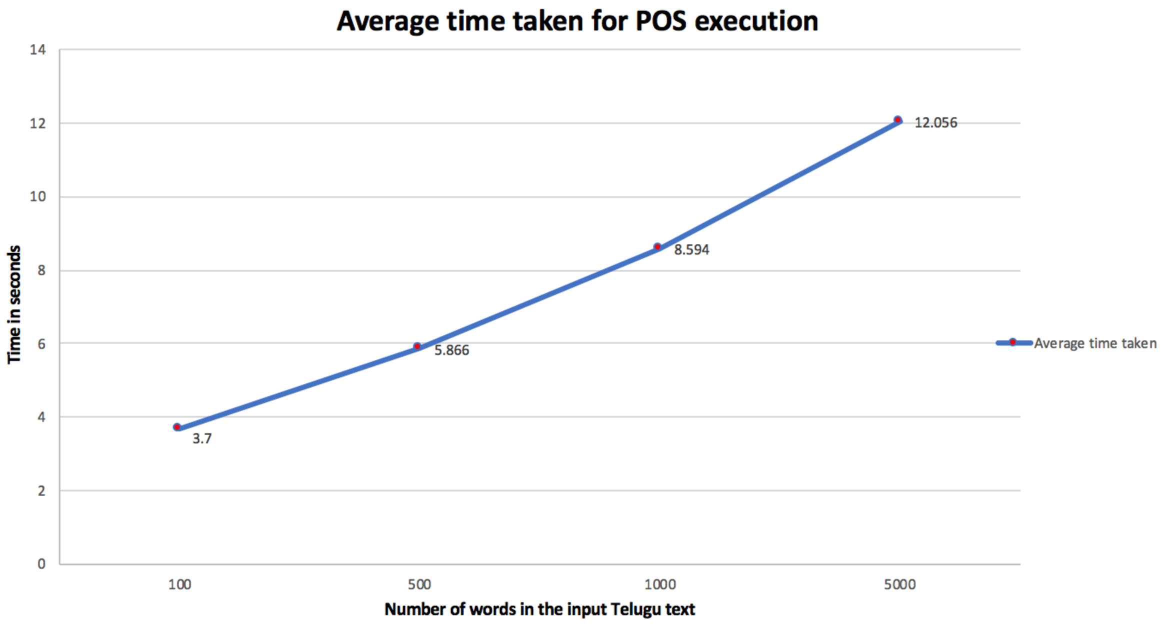


Figure 5.4: Average time taken for POS tagger execution

The average performance of Telugu SentiWordNet is shown in Table 5.7 and Figure 5.5.

Table 5.7: Average time taken for extracting polarity from Telugu SentiWordNet

Telugu SentiWordNet observations						
Number of words in a sentence	**n=1**	**n=2**	**n=3**	**n=4**	**n=5**	**Average time**
100	0.31	0.41	0.23	0.28	0.33	0.312
500	0.45	0.67	0.52	0.43	0.68	0.55
1000	0.63	0.83	0.92	0.79	1.04	0.842
5000	1.25	1.43	1.89	1.93	1.45	1.59

The overall cumulative average performance of the Telugu Sentiment Analyzer is shown in Table 5.8 and Figure 5.6.

Table 5.8: Total average time performance

Total average time observations			
Number of words in a sentence	**Average POS tagger time**	**Average Telugu Senti-WordNet time**	**Average time**
100	3.7	0.312	4.012
500	5.866	0.55	6.416
1000	8.594	0.842	9.436
5000	12.056	1.59	13.646

5.5.5 Analysis

Based on the above-average time results, it is clear that the POS system is taking a lot of time to processes the text. The derived algorithm for calculating polarity of the parsed Telugu text is yielding an effective performance and fast. The reason the low performance of POS tagger is due to the presence of huge corpus trained for multilanguage's. The overall results can be enhanced by optimizing the POS tagger or building a new POS tagger for the Telugu language.

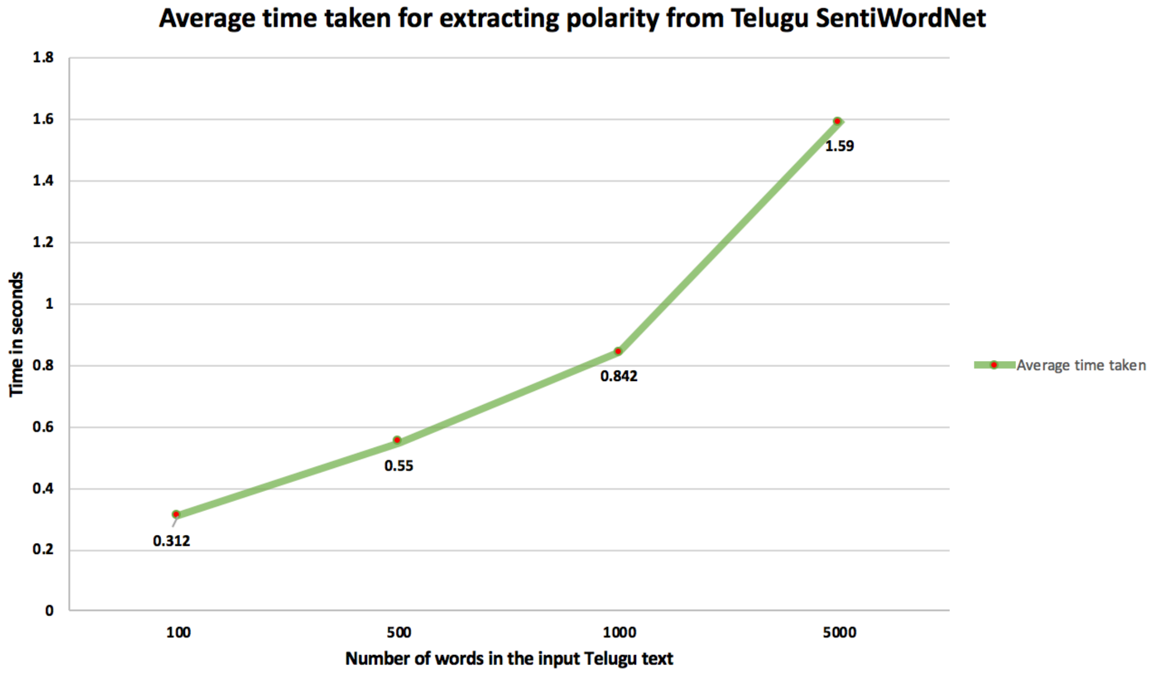


Figure 5.5: Average time taken for extracting polarity from Telugu SentiWordNet

5.6 Case Study

5.6.1 Hypothesis

Is the content in Telugu e-newspapers are biased?

Domain: Political

Description: This case study evaluates the e-newspaper articles to check if the content in the article is biased for a political party or other sectors.

5.6.2 Data collection

From all the data out online these days, it's essential to be able to identify media bias. If a journal article is biased, it means that the journalist wrote the piece has an unfair preference for somebody or something. The journalist may advocate one side of the debate or a specific politician and the reporting may be tainted. Occasionally

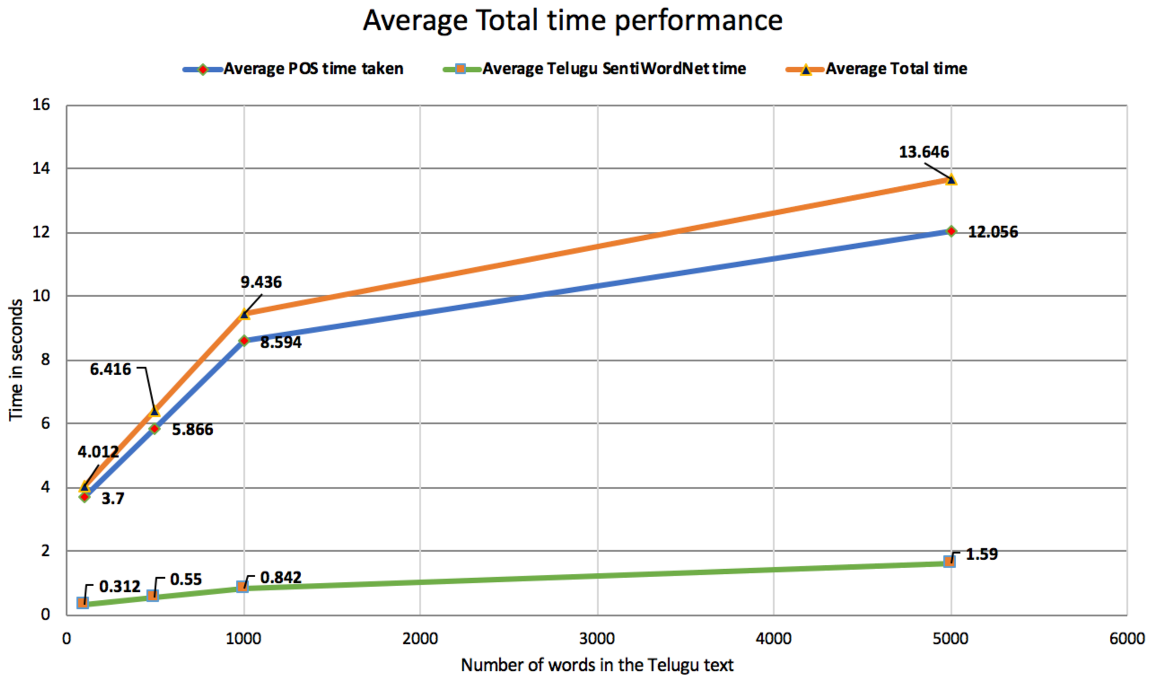


Figure 5.6: Total average time performance

journalists do not even want to be biased; they can do it by mistake or they have done insufficient studies. Sentiment analysis can help in this scenario to know the reader or users whether the article is biased or not.

As discussed previously in section 5.1, a strict criterion was followed for collecting the data considering the authorized and open accessed content. To test the stated hypothesis several assumptions were made to predict the business of e-news articles. Collected official press release statements on new schemes[7] by Government of Telangana[8] which are Telugu scripted. Initially, these government released articles were executed using the developed Telugu sentiment analysis and recorded their results. To be sure, three annotators were given a task to evaluate manually too. And these results were considered as ground-truth results.

Primary Data:

- Official press releases of new schemes from the Government of Telangana were

[7] http://rythubandhu.telangana.gov.in/
[8] https://telangana.gov.in/

Table 5.9: Primary data results (Official articles)

Name of the scheme	Number of articles	System results	Manual annotation
Raithu Bhandhu	1	Positive	Neutral
Asara Bhima	1	Neutral	Neutral
Bangaru Thalli	1	Positive	Posiive

collected.

- Data is pre-processed for removing unwanted character or symbols etc.
- Manually annotated by Telugu language experts

Secondary Data:

- Collected e-news articles based on the official scheme keywords in Telugu. E-newspapers
- Data is pre-processed for removing unwanted character or symbols etc.
- Manually annotated by Telugu language experts

5.6.3 Method

Once, the data collection and pre-processing are completed the primary data is executed over the developed prototype and sentiment polarity of them were recorded as shown in Table 5.9. Then, the secondary data obtained from Telugu e-newspapers based on the content and keywords of primary data was processed over the developed system. The results are noted and also evaluated manually by annotators as shown in the table 5.10.

5.6.4 Analysis

From the Tables 5.9 and 5.10 it clearly shows the differences between the official press releases and the editors writing. Most of the e-newspaper articles are biased in against government's new schemes to the residents of the state. News articles have a different

Table 5.10: Secondary data results (eNewspaper articles)

		System results			Manual annotation		
Name of the e-newspaper	Number of articles	Negative	Positive	Neutral	Negative	Positive	Neutral
Namaste Telangana	6	4	1	1	3	0	3
Mana Telangana	8	2	3	3	5	1	2
Velugu	4	4	0	0	4	0	0
Eenadu	12	8	1	3	10	0	3
Andhra Prabha	10	0	7	3	1	5	3
Total	40	18	12	10	23	6	11

opinion the announced official schemes. With a maximum of negative bias in the articles. This would impact user opinions on official schemes. The result from manual output is almost the same as the system results. By calculating the equation 5.1 resulted in the accuracy of 86.75%.

Chapter 6

Conclusion and Future Work

This chapter first describes the scope of the study which was described in chapter-1. It further discusses the progress of all study goals, which were attained. Finally, the chapter concludes with the possible future works and final remarks for this research work.

Reasoning draws a conclusion but does not make the conclusion certain, unless the mind discovers it by the path of experience. –Roger Bacon

This dissertation addresses the task of sentiments analysis for the Telugu text. This is a key component of other NLP tasks like responding questions tools. Most of the sentiment analysis is carried out for the dominant English language and other majorly spoken languages in the world. In the case of the Telugu, a Dravidian language there is an inadequate amount of research work. In today's internet generation, data is rapidly growing using social media, e-news articles, blogs and many more. Most of this data is unstructured but can help if that data is analyzed using NLP principles which would benefit in many ways. Sentiment analysis can help different sectors to analyze their business according to the user feedback on the digital platform by extracting sentiment from the text. Telugu is the second most spoken language in India with more than 90 million native people in the states of Telangana and Andhra Pradesh. The Telugu language is used for administration, business and education, and lots of data is exchanged in Telugu script. Hence enhancement in understanding the exchange of information has become a crucial factor. Thereby creating a vacuum of the necessity of having such tools which will give those data a better meaning and to make it easily understandable to its desired users. The core research questions that are addressed in this dissertation are:

- Question-1: What approaches can be adopted for building the Telugu Sentiment Analyzer?

  Answer: Though there are many different approaches for developing sentiment analyzers, Machine Learning and Lexicon based approach would suit the Telugu language due to its NLP limitations. In this dissertation, a lexicon-based Telugu SentiWordNet is developed to make it universal analyzer by collecting the maximum Telugu WordNet. The machine learning approach is relatively simpler and faster than SentiWordNet, but ML-based approaches are more domain-specific.

- Question-2: Does the Telugu language have enough sentiment corpus?

  Answer: As Telugu is moderately new to the digital transformations, there is not much corpus is available online, and few ML-based domain-specific datasets are available which demands a new universal corpus creation. Also, there are few projects supported by the Government of India for building a multilingual corpus for Indian languages. These projects are progressing rapidly for keeping Indian linguistics research on the global platform.

In this dissertation, initially, a comprehensive study on existing methodologies as discussed in chapter-2 and their challenges were analyzed before designing the proposed method. The analysis included the strategies of English and other majorly spoken languages sentiment analyzers. It was understood that these approaches may not be a good fit for building a Telugu sentiment analyzer because of its morphological differences along with the Subject-Object-Verb word order.

Finally, it was decided to go with Parts Of Speech (POS) based tagging as this technique results each word a POS tag. Which will be easy in classifying words in a sentence. Also, helps in identifying objective and subjective words, as objective words do not contain any sentiment and will not be useful sentiment analysis. Also, develops a new Telugu SentiWordNet for polarity and sentiment extraction.

6.0.1 Key Findings and Contributions

POS tagger

Analyzed various ways of identifying a Parts-Of-Speech tag to each word which makes the text analysis easier and faster. Lately, a European Union-funded Elexis by Sketch Engine has focused on developing multilingual text analytics systems. This system is fast and accurate in performing text analysis, it was found that this system does support Telugu language [56] and it is an open-source project. This is adopted into this dissertation for tagging the POS for the Telugu text. Though it was claimed to be fast evaluation results proved to be a bit slower in performance.

Smart filter

A smart filter was introduced to filter out the objective words based on POS tags of the words like Nouns, Pronouns which would not contain any sentiment. Also, keeps a track of the redundant words making better performance of the system. This filter was crucial in the accuracy of the developed prototype.

Telugu SentiWordNet

A lexicon SentiWordNet dictionary is a proven approach for performing universal or domain-specific sentiment analysis. The Telugu language is in its initial stage of adapting digital innovations, there was no suitable SentiWordnet was available for this language. In this dissertation, special attention was given in developing a Telugu SentiWordNet using two different approaches: (i) English-Telugu Dictionary-based and (ii) IndoWordNet based technique. The analysis and results presented in chapter-3 show that IndoWordNet based approach is beneficial for Telugu SentiWordNet that the English-Telugu Dictionary-based mechanism due to the morphological limitations.

Results

The performed experiment in chapter-4 was helpful in estimating the overall performance of the developed system. For this evaluation, several e-news articles were collected and manual annotation is done by the native speakers who are experts in the Telugu language. These results are set as ground-truth results for the error analysis.

Then the same dataset is executed on the developed system which yielded sentiment labels with polarity for each sentence. These results were counted and compared with ground-truth results. A confusion matrix is drawn from the obtained results and calculated accuracy, recall, precision, and F1-score. The accuracy of the system has resulted in 83.5%, which is promising and can be further optimized.

Time performance

Time performance is an important factor of evaluation, this decides the overall performance of the system. This estimation was split into two steps first by calculating the average time taken by the adopted POS tagger with different length of words and sentiment extraction. The time performance analysis as shown in chapter-4 was clear to understand the time performance of the system. POS tagger took a relatively longer duration when compared to the developed sentiment extraction from Telugu SentiWordNet.

Case Study

A case study was projected to check the business of the e-news articles and official government press releases using the developed system. The collected dataset is executed using the prototype designed and recorded the sentiment. To authenticate the output, it also evaluated manually by annotators who are expert in the Telugu language. It was clear that few e-news websites are biased and towards the original content and miss-leading their readers.

6.0.2 Limitations

- Annotators have their own perceptions of appropriate sentencing, the judgments of different experts regarding the same case may be inconsistent.
- Failed to analyze the Tweets because of text noise and improper syntax.
- Time performance is relatively higher due to the POS tagging.
- Does not support domain-specific analysis.
- One three polarities are examined.

- Collection of Telugu WordNet and building a Telugu SentiWordNet.
- Processing the Telugu text which was not easier due to the different syntax and research community support.

6.0.3 Future work

The research done in this study is not optimum or complete. Science advances will never stop. The field of Telugu sentiment analysis remains at a premature level. For Telugu sentiment analysis, there are many distinct fields for advancement. Improvements of NLP techniques should be initiated for the Telugu language. Telugu Dialects are of different kinds. Most have a distinctive structure and vocabulary. For each sort of Telugu dialect, the language domain requires a unique morphology tagger, a parsing tree, and a negation tagger. Telugu sentiment analysis would be further enhanced by the orientation of future actions with a fine grain sentiment corpus. Most of the work done in the Telugu sentiment analysis only took two kinds of polarity into account. In some circumstances, the strong positive, the weak, the typical positive emotion and the negative opinion must be distinguished. Need to implement this framework for other Dravidian languages to see who other languages of the same family perform the sentiment analysis. Also, building a bilingual sentiment analyzer. For future studies, numerous other options for enhancing Telugu sentiment analysis research are available. This study, for example, can further improve the results with mining techniques or models to improve the accuracy of the prediction. Second, domain-specific sentiment analysis. Plans are also to examine the efficiency of other sophisticated information retrieval systems such as BM25 and Binary Independence. The above future work is not complete, but it provides some insight into feasible potential actions. Moreover, the difficulty of Telugu in sentiment analysis as a target word makes these assignments more difficult. These obstacles should enable researchers to participate in the development of concepts to fix these issues.

Bibliography

[1] A. Bharati, R. Sangal, D. M. Sharma, and L. Bai, "Anncorra: Annotating corpora guidelines for pos and chunk annotation for indian languages," *LTRC-TR31*, pp. 1–38, 2006.

[2] Z. Wang, S. M. R. Goh, and Y. Yang, "A method and system for sentiment classification and emotion classification," Oct. 26 2017. US Patent App. 15/523,201.

[3] A. K. Mohanty, "Languages, inequality and marginalization: Implications of the double divide in indian multilingualism," *International Journal of the Sociology of Language*, vol. 2010, no. 205, pp. 131–154, 2010.

[4] L. Campbell, "Areal linguistics: A closer scrutiny," in *Linguistic Areas*, pp. 1–31, Springer, 2006.

[5] G. Lakshmeeswari, R. D. Lakshmi, and L. D. Bhaskari, "Extended encoding of telugu text for hiding compatibility," *International Journal of Computer Applications*, vol. 30, no. 5, pp. 1–7, 2011.

[6] S. G. Thomason, "Linguistic areas and language history," *Studies in Slavic and General Linguistics*, vol. 28, pp. 311–327, 2000.

[7] B. Krishnamurti, *The dravidian languages.* Cambridge University Press, 2003.

[8] T. Rama and L. Borin, *Comparative evaluation of string similarity measures for automatic language classification.*, pp. 171–200. 04 2015.

[9] J. W. Christie, "The medieval tamil-language inscriptions in southeast asia and china," *Journal of Southeast Asian Studies*, vol. 29, no. 2, pp. 239–268, 1998.

[10] S. M. Idicula and P. S. David, “A morphological processor for malayalam language,” *South Asia Research*, vol. 27, no. 2, pp. 173–186, 2007.

[11] N. Sharma, U. Pal, and F. Kimura, “Recognition of handwritten kannada numerals,” in *9th International Conference on Information Technology (ICIT’06)*, pp. 133–136, IEEE, 2006.

[12] A. K. Durga and A. Govardhan, “Ontology based text categorization-telugu document,” *International Journal of Scientific and Engineering Research*, vol. 2, no. 9, pp. 1–4, 2011.

[13] Z. Khan and T. Vorley, “Big data text analytics: an enabler of knowledge management,” *Journal of Knowledge Management*, vol. 21, no. 1, pp. 18–34, 2017.

[14] D. J. Blood and P. C. Phillips, “Recession headline news, consumer sentiment, the state of the economy and presidential popularity: A time series analysis 1989–1993,” *International Journal of Public Opinion Research*, vol. 7, no. 1, pp. 2–22, 1995.

[15] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[16] B. Liu *et al.*, “Sentiment analysis and subjectivity.,” *Handbook of natural language processing*, vol. 2, no. 2010, pp. 627–666, 2010.

[17] R. Prabowo and M. Thelwall, “Sentiment analysis: A combined approach,” *Journal of Informetrics*, vol. 3, no. 2, pp. 143–157, 2009.

[18] C. Lin and Y. He, “Joint sentiment/topic model for sentiment analysis,” in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 375–384, ACM, 2009.

[19] C. Whitelaw, N. Garg, and S. Argamon, “Using appraisal groups for sentiment analysis,” in *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 625–631, ACM, 2005.

[20] G. Vinodhini and R. Chandrasekaran, “Sentiment analysis and opinion mining: a survey,” *International Journal*, vol. 2, no. 6, pp. 282–292, 2012.

[21] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "Semeval-2016 task 4: Sentiment analysis in twitter," in *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pp. 1–18, 2016.

[22] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, vol. 50, pp. 723–762, 2014.

[23] D. J. Hughes, M. Rowe, M. Batey, and A. Lee, "A tale of two sites: Twitter vs. facebook and the personality predictors of social media usage," *Computers in Human Behavior*, vol. 28, no. 2, pp. 561 – 569, 2012.

[24] E. Javed, N. A. Mahoto, V. Khalique, and M. Ali, "Exploring role of social media usage towards academic activities of students," in *2018 5th International Multi-Topic ICT Conference (IMTIC)*, pp. 1–8, April 2018.

[25] S. Tan and J. Zhang, "An empirical study of sentiment analysis for chinese documents," *Expert Systems with applications*, vol. 34, no. 4, pp. 2622–2629, 2008.

[26] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, Association for Computational Linguistics, 2002.

[27] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2-3, pp. 103–130, 1997.

[28] Z. Niu, Z. Yin, and X. Kong, "Sentiment classification for microblog by machine learning," in *2012 Fourth International Conference on Computational and Information Sciences*, pp. 286–289, Ieee, 2012.

[29] J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *Proceedings of the ACL student research workshop*, pp. 43–48, Association for Computational Linguistics, 2005.

[30] Q. Ye, Z. Zhang, and R. Law, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches," *Expert systems with applications*, vol. 36, no. 3, pp. 6527–6535, 2009.

[31] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information Processing Management*, vol. 24, no. 5, pp. 513 – 523, 1988.

[32] Z.-H. Deng, K.-H. Luo, and H.-L. Yu, “A study of supervised term weighting scheme for sentiment analysis,” *Expert Systems with Applications*, vol. 41, no. 7, pp. 3506 – 3513, 2014.

[33] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, “Learning sentiment-specific word embedding for twitter sentiment classification,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1555–1565, 2014.

[34] A. Kennedy and D. Inkpen, “Sentiment classification of movie reviews using contextual valence shifters,” *Computational Intelligence*, vol. 22, no. 2, pp. 110–125, 2006.

[35] P. J. Stone and E. B. Hunt, “A computer approach to content analysis: studies using the general inquirer system,” in *Proceedings of the May 21-23, 1963, spring joint computer conference*, pp. 241–256, ACM, 1963.

[36] O. R. Holsti, “An adaptation of the” general inquirer” for the systematic analysis of political documents,” *Behavioral Science*, vol. 9, no. 4, p. 382, 1964.

[37] T. Wilson, J. Wiebe, and P. Hoffmann, “Recognizing contextual polarity in phrase-level sentiment analysis,” in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005.

[38] L. Deng and J. Wiebe, “Mpqa 3.0: An entity/event-level sentiment corpus,” in *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 1323–1328, 2015.

[39] K. Ahmad, C. Kearney, and S. Liu, “No news is good news: A time-varying story of how firm-specific textual sentiment drives firm-level performance,” *The European Financial Management Association*, 2013.

[40] S. Kelly and K. Ahmad, "Sentiment proxies: computing market volatility," in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 771–778, Springer, 2012.

[41] K. Ahmad, N. Daly, and V. Liston, "What is new? news media, general elections, sentiment, and named entities," in *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pp. 80–88, 2011.

[42] V. R. Mallamma and M. Hanumanthappa, "Nlp challenges for machine translation from english to indian languages," *International Journal of Computer Science and Informatics*, vol. 3, no. 01, pp. 35–40, 2013.

[43] D. Kumar and G. S. Josan, "Part of speech taggers for morphologically rich indian languages: a survey," *International Journal of Computer Applications*, vol. 6, no. 5, pp. 32–41, 2010.

[44] D. Chandramma and P. K. Pareek, "Fast and accurate parts of speech tagging for kannada-telugu pair," *International Journal of Applied Engineering Research*, vol. 13, no. 10, pp. 7857–7867, 2018.

[45] R. RamaSree and P. Kusuma Kumari, "Combining pos taggers for improved accuracy to create telugu annotated texts for information retrieval," *Dept. of Telugu Studies, Tirupathi, India*, 2007.

[46] S. T. Eckmann, G. Vigna, and R. A. Kemmerer, "Statl: An attack language for state-based intrusion detection," *Journal of computer security*, vol. 10, no. 1-2, pp. 71–103, 2002.

[47] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proceedings of the 25th international conference on world wide web*, pp. 145–153, International World Wide Web Conferences Steering Committee, 2016.

[48] K. Buryak, A. Swerdlow, C. Roux, L. H. Swartz, and C. Johny, "Cluster-based language detection," Aug. 11 2015. US Patent 9,104,744.

[49] J. Brooke and G. Hirst, "Native language detection with 'cheap'learner corpora," in *Twenty Years of Learner Corpus Research. Looking Back, Moving Ahead: Proceedings of the First Learner Corpus Research Conference (LCR 2011)*, vol. 1, p. 37, Presses universitaires de Louvain, 2013.

[50] L. Ramasamy, Z. Žabokrtskỳ, and S. Vajjala, "The study of effect of length in morphological segmentation of agglutinative languages," in *Proceedings of the First Workshop on Multilingual Modeling*, pp. 18–24, Association for Computational Linguistics, 2012.

[51] K. Sunitha and N. Kalyani, "A novel approach to improve rule based telugu morphological analyzer," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 1649–1652, IEEE, 2009.

[52] N. Habash and O. Rambow, "Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop," in *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)*, pp. 573–580, 2005.

[53] L. Màrquez and H. Rodríguez, "Part-of-speech tagging using decision trees," in *European Conference on Machine Learning*, pp. 25–36, Springer, 1998.

[54] Y. Vyas, S. Gella, J. Sharma, K. Bali, and M. Choudhury, "Pos tagging of english-hindi code-mixed social media content," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 974–979, 2014.

[55] V. Singh and B. Saini, "An effective tokenization algorithm for information retrieval systems," *Departement of Computer Engineering, National Institute of Technology Kurukshetra, Haryana, India*, 2014.

[56] S. Reddy and S. Sharoff, "Cross language pos taggers (and other tools) for indian languages: An experiment with kannada using telugu resources," in *Proceedings of the Fifth International Workshop On Cross Lingual Information Access*, (Chiang Mai, Thailand), pp. 11–19, Asian Federation of Natural Language Processing, November 2011.

[57] A. Das, "Opinion extraction and summarization from text documents in bengali," *Kolkata, India*, 2011.

[58] C. Fellbaum, "Wordnet," in *Theory and applications of ontology: computer applications*, pp. 231–243, Springer, 2010.

[59] K. J. Millman and M. Aivazis, "Python for scientists and engineers," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 9–12, 2011.

[60] M. F. Sanner *et al.*, "Python: a programming language for software integration and development," *J Mol Graph Model*, vol. 17, no. 1, pp. 57–61, 1999.

[61] M. Beyreuther, R. Barsch, L. Krischer, T. Megies, Y. Behr, and J. Wassermann, "Obspy: A python toolbox for seismology," *Seismological Research Letters*, vol. 81, no. 3, pp. 530–533, 2010.

[62] A. Nagpal and G. Gabrani, "Python for data analytics, scientific and technical applications," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pp. 140–145, Feb 2019.

[63] S. Taneja and P. R. Gupta, "Python as a tool for web server application development," *International Journal of Information Communication and Computing Technology*, vol. 2, no. 1, pp. 77–83, 2014.

[64] S. I. Feldman, "Make—a program for maintaining computer programs," *Software: Practice and experience*, vol. 9, no. 4, pp. 255–265, 1979.

[65] D. H. Martin and J. R. Cordy, "On the maintenance complexity of makefiles," in *Proceedings of the 7th International Workshop on Emerging Trends in Software Metrics*, WETSoM '16, (New York, NY, USA), pp. 50–56, ACM, 2016.

[66] S. Aman and S. Szpakowicz, "Identifying expressions of emotion in text," in *Text, Speech and Dialogue* (V. Matoušek and P. Mautner, eds.), (Berlin, Heidelberg), pp. 196–205, Springer Berlin Heidelberg, 2007.

[67] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[68] K. M. Ting, “Confusion matrix,” *Encyclopedia of Machine Learning and Data Mining*, pp. 260–260, 2017.

[69] T. C. Landgrebe, P. Paclik, and R. P. Duin, “Precision-recall operating characteristic (p-roc) curves in imprecise environments,” in *null*, pp. 123–127, IEEE, 2006.

Appendix

Table 1: Key for Telugu POS

Tag type	Number of tags	Tags (refer Appendix Table-2 and Table-3)
Main POS Tag	25	JJ (Adjective), NN (Noun), VM (Verb-finite), . .
Coarse POS Tag	9	n (noun), num (number), unk (unknown). . .
Gender	6	any, f (female), m (male),null
Number	4	any, pl (plural), sg (singular), null
Person	5	1, 2, 3, any, null

Table 2: POS Tag Set for Indian Languages (IIIT Hyderabad)[1]

Sl No.	Category	Tag name	Example
1.1	Noun	NN	
1.2	NLoc	NST	
2.	Proper Noun	NNP	
3.1	Pronoun	PRP	
3.2	Demonstrative	DEM	
4	Verb-finite	VM	
5	Verb Aux	VAUX	
6	Adjective	JJ	
7	Adverb	RB	*Only manner adverb
8	Post position	PSP	
9	Particles	RP	bhI, to, hI, jI, hA.N, na,
10	Conjuncts	CC	bole (Bangla)
11	Question Words	WQ	
12.1	Quantifiers	QF	bahut, tho.DA, kam (Hindi)
12.2	Cardinal	QC	
12.3	Ordinal	QO	
12.4	Classifier	CL	
13	Intensifier	INTF	
14	Interjection	INJ	
15	Negation	NEG	
16	Quotative	UT	ani (Telugu), endru (Tamil), bole/mAne (Bangla), mhaNaje (Marathi), mAne (Hindi)
17	Sym	SYM	
18	Compounds	*C	
19	Reduplicative	RDP	
20	Echo	ECH	
21	Unknown	UNK	

Table 3: Chunk Tag Set for Indian Languages[1]

Sl. No	Chunk Type	Tag Name	Example
1	Noun Chunk	NP	Hindi: ((merA nayA ghara))_NP “my new house”
2.1	Finite Verb Chunk	VGF	Hindi: mEMne ghara para khAnA ((khAyA_VM))_VGF
2.2	Non-finite Verb Chunk	VGNF	Hindi: mEMne ((khAte – khAte_VM))_VGNF ghode ko dekhA
2.3	Infinitival Verb Chunk	VGINF	Bangla : bindu Borabela ((snAna karawe))_VGINF BAlobAse
2.4	Verb Chunk (Gerund)	VGNN	Hindi: mujhe rAta meM ((khAnA_VM))_VGNN acchA lagatA hai
3	Adjectival Chunk	JJP	Hindi: vaha laDaZkI hE((suMdara_JJ sI_RP))_JJP
4	Adverb Chunk	RBP	Hindi : vaha ((dhIre- dhIre_RB))_RBP cala rahA thA
5	Chunk for Negatives	NEGP	Hindi: ((binA))_NEGP ((kucha))_NP ((bole))_VG ((kAma))_NP ((nahIM calatA))_VG
6	Conjuncts	CCP	Hindi: ((rAma))_NP ((Ora))_CCP ((SyAma))_NP
7	Chunk Fragments	FRAGP	Hindi; rAma (jo merA baDZA bhAI hE) ne kahA...
8	Miscellaneous	BLK	