**CHAPTER 1**

**INTRODUCTION**

In this chapter, the concept of Congruent Networks is discussed. After the introduction to the domain, the challenges and opportunities that existed during the implementation of this project are stated. We then present the problems that have been identified for this domain. For the problems that have been identified, we have selected a few problems for our implementation of this project. We then describe all the problems for which we have attempted to provide a solution.

* 1. **Introduction to Congruent Networks**

According to the fundamental theory of Social behavior of humans, we create our own social network. We form a network of our ‘friends’. A ‘friend’ does not mean a classmate or a colleague. A ‘friend’ in this context, means a person who is involved with us in some social behavior. A social link can be established based on various parameters such as hobbies, folksonomy, profile, views etc. These networks have a lifetime, which is usually long, and exists till we have at least one friend. As our social behavior changes, the friends in our network will change, and this will result in social networks being dynamic. A network will be strong when the right ‘friends’ are connected. This is possible only in a context where the friends are tied together, stronger in a circumstance to perform a task.

The force that ties us stronger in a network can be called as “Congruence” and such networks can be called as “Congruent Networks”. The general definition of congruence is: identical in form. in agreement or harmony. A feeling of inherent trust comes from someone who is congruent. Whether intentional or not, congruence often attracts one to another person. People seek friends who conduct themselves congruently in their presence. Students get inspiration from teachers who are congruent with them.

* 1. **Egocentric Congruent Networks**

Egocentrism is found in all people, but the extent of ego can vary from person to person, and it also depends on the context. Egocentric people see themselves and their own interests or opinions as most important. Decisions that are taken based on one’s own information, is regarded to be more important than the decisions that are taken with other people’s information as well.

Although much social network analysis attempts to examine complete, large scale networks (ones in which every connection between network members is visible), we consider a more

restrictive view of using egocentric network analysis technique, which examines only people’s immediate neighbors and associated interconnections. This helps us to learn about how individuals correspond with their own social networks.

An ego network is a mini-network or immediate neighborhood surrounding ego that can, perhaps, reveal something important about the social world from the ego’s perspective. We might guess, for example, that if the social units are persons, those in the ego network are the ones that have the greatest impact on ego’s attitudes, norms, values, goals and perceptions of the world. Moreover, those are the ones to whom ego must turn to seek information, help and support. In some sense, an ego network is a map of ego’s personal social world. It shows something about how that ego is tied in to the larger human society.

* 1. **Challenges and Opportunities**

The profile of a user can keep changing as the user’s social actions changes. Some social networking sites are providing services which keeps the users to stick onto the sites. The actions made by users on the social network sites indicate their presence in the online world. Nowadays, users are trying to increase their Social Index**,** which means that their availability on social networking sites is more frequent and they like to be connected socially**.**

Current electronic social networks provide very few services that utilize the Social Index of the user effectively. They are mainly targeting the users as commercial entities, with a lot of targeted advertisements. These services, to the extent are providing one way benefits to the commercial users (with advertisements). Many social networking sites are available today in the market, each providing a particular set of services. Thus there is a need to create some kind of a network which is capable of providing the aggregated and combined services, with mutual benefits to the users (regular) as well as the commercial entities. The current networking sites are ineffective in capturing **“contexts”**. A context is some circumstance which provides mutual benefits to the stakeholders.

Different people perceive ego in different ways. Some people are more egocentric, and are more worried about what benefits they reap, whereas some people are more sociocentric, and are more concerned about the society as a whole, and not just themselves. The main challenge is the way

in which we are able to capture this ego, and how we match the ego of one person with that of another person. The other main challenge is to generate the egocentric congruent network for the context that is given by the user.

The user may not be in a position to express the context as a single word or a phrase. It is easier for the user to provide a simple description of the context, and let us decide on what context the user is in. Thus the challenge here is to provide a natural language interface to the user, through which the user can describe his context. If the context of the user is already known to us, we can also recommend some possible contexts, out of which the user can select his context. Through the same interface, the user can also add in some properties to the context of the user.

The next challenge here is to measure the degree of an egocentric congruent network is the extent to which the nodes in the network, are congruent with each other. The information of each user is expressed as his profile, and based on the profiles of the users in the congruent network, the degree of the egocentric congruent is calculated. This degree is a measure of how closely related are the people in the network.

The other challenge here is to measure the stability of the egocentric congruent network is a measure of the number of nodes in the network that may not be congruent throughout the entire lifetime of the context. Each node in the network also has a lifetime, and the node should exist in that context, throughout the lifetime of the context. The number of nodes that leave or join the congruent network, or are becoming incongruent with the context, is a measure of the stability of the congruent network.

* 1. **Problems Identified**

After analyzing the introduction to congruent networks, and the challenges and opportunities that existed during the implementation of this project, we list a few problems which we have indentified:

* + - Creation of Egocentric Networks
    - Handling the perception of ego by different people
    - Creation of context
    - Capturing the ego of a person
    - Matching the ego between people
    - Matching the ego of a person with the context in which he is in
    - Creation of Egocentric Congruent Networks for the given context
    - Measuring the degree and stability of the Egocentric Congruent Network
  1. **Applications**

Some areas where egocentric congruent networks are applicable are mentioned below:

* + - Matrimonial services
    - Human relationship management
    - Team formation
    - Team productivity
    - Friend suggestion

**CHAPTER 2**

**PROBLEM DEFINITION**

Flamingo is a stylistic analyser. Word processors like MS Word provide syntax and semantics functionalities like spell check and grammar checks. However, this application evaluates a document against rules of style. Stylistics is a field of study that focuses on interpretation of texts in regard to their linguistic and tonal style. Linguistics is the study of language. So, this application attempts on enhancing the stylistic aspect of a piece of text thus making it more pleasing to read. The purpose is to benefit writers of essays, articles or even short stories by providing stylistic comments. The concept of a stylistic analyser is a novel idea attempting to better the style of writing.

**2.1** **Stylistic evaluation of text**

Initially, we create a compilation of various rules of style. These rules of style are from popular and widely acclaimed books like *The Elements of Style* by William Strunk Jr. and E. B. White. This book puts forth several rules that, if followed, gives rise to crisp, clear and well-organized writing. Flamingo uses a select set of rules from this book.

This application intends on evaluating a document for non-conformance of these rules of style. They appear as comments that user may choose to ignore but if corrected will better the piece of text. Certain aspects are so deeply embedded in our writing style like usage of passive voice or construction of long winding sentences with plenty of descriptive words. The application catches such seemingly trifling “errors” that can be hindrances to the reader of that text.

**2.2 Evaluation of readability**

Readability is the ease with which text can be read and understood. There are several measures of readability that associate the readability of a text with a number. Some measures like Gunning-Fog index and SMOG index give a formulae denoting the number of years of formal education required to easily read a piece of text. Some others like the Flesch-Kincaid Reading Ease evaluate to a score from 0 to 100 - the higher the score, the easier is the text.

Flamingo, for a given text, calculates several readability scores and gives an average value. This gives the user an idea as to which age group his/her document caters. It can also help the user write keeping his target audience in mind. The application can be configured to alert the user when his writing goes either above or below the readability range for the target audience. Readability may be evaluated as and when the user requests for it or dynamically – as the user types.

**2.3 Floweriness and obscurity**

[Lit survey?] Floweriness is a measure of how elaborate a piece of text is. It is the number of adjectives and adverbs per sentence. The higher this number, the greater is the floweriness. For an article or an essay, a less flowery text is preferred. However, works of literature may have a high floweriness count.

Obscurity is a measure of the complexity of a piece of text, in terms of the number of rare words. A word is rare it its frequency of usage in the English language is very low. Obscurity is the number of rare words used per sentence. As is the case with floweriness, each user’s requirement of obscurity may be different based on the type of text he is writing.

This application provides the floweriness and obscurity count of a piece of text. The measure is provided as and when the user requests for it. The user may interpret the values of the measures according to the type of document he is writing. A literary work of art may be very flowery with the usage of complex and infrequently used words. A scientific article or essay must be less flowery but may still have a large set of rarely used words. However, a composition for children will necessarily be less descriptive with simple words.

**2.4 Adherence to a template**

When the user wishes to write a piece of text that requires a particular format, he may waste time searching for a suitable template. Flamingo provides templates of popular formats like school essay and formal letter. The user can fill the appropriate data according to the template provided. After completion, the user can view his document with the text provided appended with help text. The user may also edit it further. Conformance to the template is not checked when the user begins editing his document.

**2.5 Simple text editor**

At the heart of this plethora of functionality is that Flamingo is a text editor. It provides the user all the basic features of a text editor – new, open, undo, redo, cut, copy and paste. The application can save text only in a .txt format file, but can read any file with ASCII characters. Just like other text editors, Flamingo does not provide text-formatting functionality.

**CHAPTER 3**

**LITERATURE SURVEY**

**3.1 Survey experience**

3.1.1 Social networks

A social networking service is a platform to build [social networks](http://en.wikipedia.org/wiki/Social_network) or [social relations](http://en.wikipedia.org/wiki/Social_relation) among people who may share interests, activities, backgrounds, or real-life connections. A social network service consists of a representation of each user (often a profile), his/her social links, and a variety of additional services. Most social network services are [web-based](http://en.wikipedia.org/wiki/Web-based) and provide means for users to interact over the [Internet](http://en.wikipedia.org/wiki/Internet), such as [e-mail](http://en.wikipedia.org/wiki/E-mail) and [instant messaging](http://en.wikipedia.org/wiki/Instant_messaging). [Online community](http://en.wikipedia.org/wiki/Online_community) services are sometimes considered as a social network service, though in a broader sense, social network service usually means an individual-centered service whereas [online community](http://en.wikipedia.org/wiki/Online_community) services are group-centered. Social networking sites allow users to share ideas, pictures, posts, activities, events, and interests with people in their network.

Web-based social networking services make it possible to connect people who share interests and activities across political, economic, and geographic borders. Several websites are beginning to tap into the power of the social networking model for [philanthropy](http://en.wikipedia.org/wiki/Philanthropy). They provide a means for connecting otherwise fragmented industries and small organizations without the resources to reach a broader audience with interested users. Social networks are providing a different way for individuals to communicate digitally. These communities of hypertexts allow for the sharing of information and ideas, an old concept placed in a digital environment.

The advent of [social networking](http://en.wikipedia.org/wiki/Social_networking) platforms may also be impacting the way(s) in which learners engage with technology in general. The use of online social networks by school libraries is also increasingly prevalent and they are being used to communicate with potential library users, as well as extending the services provided by individual school libraries. Professional use of social networking services refers to the employment of a network site to connect with other professionals within a given field of interest. Social networking services like [LinkedIn](http://en.wikipedia.org/wiki/LinkedIn) geared towards companies and industry professionals looking to make new business contacts or keep in touch with previous co-workers, affiliates, and clients. Other network sites are now being used in this manner, [Twitter](http://en.wikipedia.org/wiki/Twitter) has become a mainstay for professional development as well as promotion and online Social networking services support both the maintenance of existing social ties and the formation of new connections.

Much of the early research on online communities assume that individuals using these systems would be connecting with others outside their preexisting social group or location, liberating them to form communities around shared interests, as opposed to shared geography. Other researchers have suggested that the professional use of network sites produce “[social capital](http://en.wikipedia.org/wiki/Social_capital).” For individuals, social capital allows a person to draw on resources from other members of the networks to which he or she belongs. These resources can take the form of useful information, personal relationships, or the capacity to organize groups. As well, networks within these services also can be established or built by joining special interest groups that others have made, or creating one and asking others to join.

Social networks exhibit strikingly systematic patterns across a wide range of human contexts. Although genetic variation accounts for a significant portion of the variation in many complex social behaviors, the heritability of egocentric social network attributes is unknown.

3.1.2 Social network analysis

The roots of social network analysis are found in the mathematical study of graph theory (such as the work of Erdos, Harary and Rappaport) and empirical studies of social psychology (Bott, Heider and Moreno). While the former group were charting various axioms between abstract nodes and lines, the latter found nodes and lines to be a sensible way to map concrete relationships between individuals. As the field matured in the latter half of the twentieth century these two groups converged on a series of metrics and methods to tease out underlying structures from complex empirical phenomena.

Social network analysis is a rapidly expanding interdisciplinary paradigm, much of which is taking place with online data. Social network analysis has emerged in the past half-century as a compelling complement to the standard toolkit of social science researchers. At its foundation is a belief that explanations for social organization are not to be found in innate drives or abstract forces. Instead we can look to the structure of relationships that constrain and enable interaction alongside the behaviors of agents that reproduce and alter these structures. Social network analysis is particularly well suited to understanding online interaction. There are two key facts about online interaction that make it particularly amenable to social network analysis - the nature of online interaction and the nature of digital information.

We can learn a lot about people from how they talk, to whether they talk, to sociable, well connected people or to unconnected individuals. Online communication is an increasingly important way to get information from, and keep in contact with, each other. Communication through a computer can be recorded and then analyzed.

Network analysis began to mature in the 1970s. In 1969, Stanley Milgram published his Small World experiment, demonstrating the “six degrees of separation”. In 1973, Mark Granovetter’s published the landmark “The Strength of Weak Ties” which showed empirically and theoretically how the logic of relationship formation led to clusters of individuals with common knowledge and important ’weak tie’ links between these clusters. The following two decades saw explosive growth in the number of studies that either alluded to or directly employed network analysis.

One of the most basic notions governing the structure of social networks is homophily, the principle that we tend to be similar to our friends. (or the tendency for people to choose relationships with people who have similar attributes). Homophily provides a fundamental illustration of how a network’s surrounding contexts can drive the formation of its links. Nodes are connected in the network depending on the several characteristics, these can be listed as follows

* Nodes in the network are similar to you along racial and ethnic dimensions
* similar in age
* similar in characteristics that are more or less mutable
* places they live
* occupations
* their levels of affluence
* their interests
* beliefs,
* opinions
* relationship

3.1.3 Communities of Interest

Communities of interest are groups of people who share a common identity, or a group of people who share a common experience. Hence people who share some common interests can form a community of interest.  Participation in a community of interest can be compelling, entertaining and create a ‘sticky’ community where people return frequently and remain for extended periods.

3.1.4 Communities of Purpose

A community of purpose is a [community](http://en.wikipedia.org/wiki/Community) of people who are going through the same process or are trying to reach a similar goal. Such communities serve a functional purpose, smoothing the path of the member for a limited period surrounding a given activity. Members of the community assist each other by sharing experiences, suggesting strategies and exchanging information on the process in hand.

3.1.5 Connection maps by LinkedIn - InMaps

InMaps are used to visualize how your network looks like. It would be interesting to find out whether one’s connections form clusters or groups. It would be nice if one could see the way all connections are related to each other and also be able to identify the elusive hubs between one’s professional worlds. InMaps are capable of doing this.

3.1.6 Ego networks

Ego networks have a simple but constrained structure, which makes it simple for us to collect data, but we will have fewer powerful tools with which to analyze the data. Techniques that have been used to analyze ego networks are usually based on density, connectivity or the attributes of the neighboring nodes, or combinations of all the three.

3.1.7 Ego network analysis

Ego network analysis focuses on network characteristics such as size, relationship strength, density, [centrality](http://en.wikipedia.org/wiki/Centrality), [prestige](http://en.wiktionary.org/wiki/prestige) and roles such as [isolates, liaisons](http://en.wikipedia.org/wiki/Isolates), and [bridges](http://en.wikipedia.org/wiki/Bridge_(interpersonal)). Such analyses, are most commonly used in the fields of [psychology](http://en.wikipedia.org/wiki/Psychology) or [social psychology](http://en.wikipedia.org/wiki/Social_psychology_(sociology)), [ethnographic](http://en.wikipedia.org/wiki/Ethnographic) [kinship](http://en.wikipedia.org/wiki/Kinship) analysis or other [genealogical](http://en.wikipedia.org/wiki/Genealogy) studies of relationships between individuals.

3.1.8 Interest graphs

An interest graph is an online representation of the specific and varied things in which an individual is interested. Interest graphs have perceived utility and value because of the premise that people’s interests are a major aspect of who they are, forming part of their personal identity, and can be used as indicators of such things as what they might want to do or buy, where they might want to go, or who they might want to meet, follow or vote for.

3.1.9 Network visualization techniques

Visual representation of social networks is important to understand the network data and convey the result of the analysis. Many of the analytic software have modules for network visualization. Exploration of the data is done through displaying nodes and ties in various layouts, and attributing colors, size and other advanced properties to nodes. Visual representations of networks may be a powerful method for conveying complex information, but care should be taken in interpreting node and graph properties from visual displays alone, as they may misrepresent structural properties better captured through quantitative analyses.

3.1.10 Collection of user profile information from existing social networks

The main focus of this project is to identify the users of a social network who are closely related to each other in some context. This would mean that the information of every user in the social network has to be collected. This is the first step in order to generate a congruent network of people. For this project, the information of users has been collected from Facebook using the Graph API. The Graph API is the primary way to collect data from the [social graph](https://developers.facebook.com/docs/opengraph/) of Facebook. It is a low-level HTTP-based API that can be used to query data and post new stories.

In order to collect a user’s information from Facebook using the Graph API, the access token of the user has to be obtained. The access token is a random string that identifies a user and provides information about granted permissions. Once the user access token is obtained, we can collect the information of the user, as well as the information of the user’s friends on Facebook, using a HTTP request.

**3.2 Focused literature survey**

A social network is the connection of people by a computer network, and internet will be the larger social network. The main point is that a social network is composed of actors and ties between them. Ties in a social network are numerous and of different natures: familial ties, lifelong friend ties, marital ties, business partner ties, that are important for people to obtain the fundamentals of identity, affection, emotional and material support, i.e. the recognition of their existence by others. However, the commitment of individuals is superficial, limited to the reasons of the local interaction. As a consequence, ties in a social network are "socially-oriented" like in the real life, trust does not play an important role, and individuals generally belong to several social networks

[Social networking](http://personalweb.about.com/od/easyblogsandwebpages/ss/2007topsnsites.htm) is a way for one person to meet up with other people on the internet. That's not all though. Some people use social networking sites for meeting new friends on the internet. Other's use it to find old friends. Then there are those who use it to find people who have the same problems or interests they have.

We can learn a lot about people from who they talk to — whether they talk to sociable, well connected people or to unconnected individuals, for example. Is the person part of a large group whose members all talk to each other, or does he or she bridge social worlds? Examining such connections between people can teach us how they operate socially.

Online communication, in particular, is an increasingly important way to get information from, and keep in contact with, each other. Communication through a computer can be recorded and then analyzed; we can later use that analysis to develop systems that are more aware of how people interact online. Although much social network analysis attempts to examine complete, large scale networks, ones in which every connection between network members is visible, we take a more restrictive view. Using egocentric network analysis techniques, which examine only people’s immediate neighbors and associated interconnections, helps us learn about how individuals correspond with their social networks.

3.2.1 Functions Served by Ego Networks

Ego networks mainly serve the following purposes:

* Social support
* Emotional and material aid
* Companionship
* Information.
* Sense-making
* How to interpret the world
* Social control
* Ensuring that egos behave according to norms
* Access to resources
* Entrepreneurs draw on their contacts to get the clients and employees and consultants they need
* Behavioral models
* You tend to talk like the people you talk to
* You become aware of choices from the people in your ego network

By an egocentric network, we mean one that represents data about the relationships of a single person in respect to others.

* + 1. Social network analysis

Social network analysis offers a powerful framework for detecting and interpreting social relationships online. They are accompanied by a host of analytic techniques ranging from simple centrality scores to sophisticated multilevel modeling. Yet gathering these networks is a time-intensive and challenging task. Online networks make this task somewhat easier through the use of passive networks (such as email stores and web pages), but the increase in efficiency leads to additional challenges about when to stop collecting, and what sorts of relations are substantively meaningful. Overcoming these challenges takes patience, a good dose of technical skills with scripting languages or custom software and some trial and error. In return the results, as seen by many of the aforementioned studies, can inform our understanding of the interpersonal structures that affect online participation and online life in general. Yet, the techniques are relevant beyond the digital domain. The discovered structures mirror and are a part of everyday life. It is not merely a gaze to distant shores, but a more crystallized view to the here and now.

* + 1. Context in a social network

There is a large number of social networks. Depending on whom you are communicating with and how you want your message to be received by the recipient(s), you would choose your tool from a lot of choices. Implementing and analyzing a context is also another challenge. Not many social networks have been successfully able to bring the context into consideration.

* + 1. Ego networks

In analysis the entire network, the goal is often to describe the characteristics of the network, and ask why certain individuals occupy a particular location in the network. (E.g., why do people always reply to him? Are there multiple subgroups in this network?) By contrast, ego network analysis is comparative in nature. One examines the differences in the size, shape and quality of a number of personal networks or ego networks. These networks are commonly captured by sampling from a population. In this regard they are akin to traditional surveys as one would similarly want a representative (even stratified) random sample from a population. Each sampled case in this context is referred to as “ego”, and the nodes connected to ego are referred to as “alters”. One can either capture a star network (which is merely the ties to ego) or a full personal network (which includes the ties between alters). One can unobtrusively collect personal networks in social software sites, communication and web pages. In each case one captures a list (such as a friend list) and then checks to see who on this list is also tied to each other. Active collection of personal networks can make use of a number of pre-existing interview and survey techniques.

**CHAPTER 4**

**PROJECT REQUIREMENTS DEFINITION**

**4.1 Project Perspective**

The user in order to use the application must start with either a blank document or open an existing file in the editor window. The text in the window can then be evaluated for its floweriness, obscurity and readability.

In addition to that, numerous rules of style are checked. The rules are a combination from the book “The Elements of Style” by Strunk and White and some universally accepted guidelines for writing appropriately. These are displayed in forms of suggestions and comments to the user which he/she can choose to ignore or act upon.

Some of the rules include an appropriate length of a sentence, discouraging use of passive voice and ending a sentence with a preposition, tense inconsistency in a sentence, etc.

It also includes features for detection/correction of commonly misused words in English, and some misused phrases in Indian English. The comments for these also suggests alternatives for the correction of these phrases

**4.2 Project Functions**

The major facilities that are provided by our system are the following:

* Evaluation of readability of a document
* Templates for various formats such as Debate, School Essay and Formal Letter
* Evaluation of the floweriness of the document
* Evaluation of the obscurity of the document
* Detecting very long sentences
* Detecting the use of passive voice
* Detecting and providing alternative to misused words
* Detecting and providing alternative to misused phrases in Indian English
* Detecting tense inconsistency
* Checking if a sentence ends with a preposition
* Typing text into the editor window, with advanced features like copy-pasting and cut-pasting
* Checking if the readability matches the target readability or not
* Creation, Opening and saving a text file

**4.3 User Classes and Characteristics**

The user can open a text file in the editor window or type directly and save the file later. They can choose the purpose of the document and adhere to the template provided for the same.

They can get their readability scores by clicking on Calculate Readability option under Tools menu or by pressing Ctrl+G. They can also choose readability to be evaluated dynamically by clicking the Auto-evaluate readability option under the Tools menu.

They can choose to evaluate the text against the set rules of style by clicking the Evaluate Readability option under Tools menu. When they click on the comment, the sentence for which the comment was generated gets highlighted. The comments are displayed in different colours based on their severity. The user can choose which set of comments to display by clicking the toggle buttons above the comments section.

* 1. **Operating Environment**

4.4.1 Hardware requirements

* Processor : Intel Core i3 and above
* RAM : 2GB and above
* Hard disk : 20GB and above

4.4.2 Software requirements

* Operating System : 64 bit Windows or linux
* Language : Python 2.7
* Python packages : NLTK
* UI: PyQT

**4.5 Assumptions and Dependencies**

The biggest constraints for this project are the limitations of the Natural Language processing subsystem, in this case NLTK. The system is only as good as the parser it uses; hence any wrong tagging/tokenizing/parsing on NLTK’s front shall also result in wrong output in the system. An assumption for this system is that the user is familiar with the English language.

.

**CHAPTER 5**

**CHAPTER 5**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**5.1 External Interface Requirements**

5.1.1 User Interface

* Interface for typing the document directly – Editor Window
* Interface for saving the file – Save Dialogue Box
* Interface for entering text into Debate format – Debate Dialogue Box
* Interface for entering text into Formal Letter format – Formal Letter Dialogue Box
* Interface for entering text into School Essay format – School Essay Dialogue Box
* Interface for viewing generated statistics and comments – Editor Window upon pressing/clicking the relevant buttons/options.

5.1.2 Software Interfaces

* Operating system : 64–bit linux 2.4 or 64-bit windows
* Language : Python
* NLTK package for python
* PyQT version 4.10.4

**5.2 Functional Requirements**

5.2.1 Applying rules from Elements of Style and giving suggestions

5.2.1.1 Introduction: The software should be able to detect wrong usage, common errors, etc. based on the rules mentioned in Elements Of Style and give suggestions accordingly.

5.2.1.2 Input: The text being typed by the user or an existing text document will be the input for the software.

5.2.1.3 Processing: The software will analyse the text after every few minutes and check for its conformance to the rules of style. If the text does not follow these rules, the software will generate comments associated with the respective line numbers for which the comment is meant. It will also give suggestions for such cases, where it is feasible.

5.2.1.4 Output: The output will be a set of comments generated by the software based on the text written and its conformance to the rules of style.

5.2.1.5 Error Handling: The software will ensure that it doesn’t crash for invalid input. Languages and scripts other than English and Alphabet will be ignored so as to not generate comments that are not applicable for the text.

5.2.2 Evaluating the readability score of the document

5.2.2.1 Introduction: The text will also be analyzed for its readability and check if it matches the target readability of the document.

5.2.2.2 Input: The text being typed by the user or an existing text document along with the target readability will be the input for the software.

5.2.2.3 Processing: Based on different formulae for readability score, the software will analyze the text, and generate its readability score. This will then be matched with the target readability score and if the difference is huge, a comment will be generated to inform the user of the same.

5.2.2.4 Output: Various readability scores of the document. Comments if any of it doesn’t match the target readability.

5.2.2.5 Error Handling: Since the readability test is evaluated for American English, words that are not found in the CMU Dictionary are ignored.

5.2.3 Giving Suggestions to make the text conform to the target writing style

5.2.3.1 Introduction: If the user has selected a particular writing style, the software will check its conformance to the writing style chosen and give suggestion as the user types.

5.2.3.2 Input: The text being typed by the user or an existing text document, along with the style of writing chosen for the document will be the input for the software.

5.2.3.3 Processing: Different rules will be hardcoded for each style. The software will analyze the text based on the rules of the writing style chosen. Comments will be generated accordingly.

5.2.3.4 Output: The output will be a set of comments generated by the software based on the text written and its conformance to the rules of the writing style chosen by the user.

5.2.4 Evaluating the floweriness of the text

5.2.4.1 Introduction: The software will analyse the text for its floweriness.

5.2.4.2 Input: The text being typed by the user or an existing text document will be the input for the software.

5.2.4.3 Processing: Based on the number of adjectives and adverbs present in the text, the software will evaluate its floweriness. Based on that, comments will be generated.

5.2.4.4 Output: Comments about the floweriness of the text.

5.2.4.5 Error Handling: Languages and scripts other than English and Alphabet will be ignored so as to not generate comments that are not applicable for the text.

5.2.5 Evaluating the obscurity of the text

5.2.4.1 Introduction: The software will analyse the text for its obscurity.

5.2.4.2 Input: The text being typed by the user or an existing text document will be the input for the software.

5.2.4.3 Processing: Based on the frequency of each word in the text, the total number of rare/difficult words will be calculated. This number, divided by the total number of sentences, will be displayed as the obscurity of the text

5.2.4.4 Output: Comments about the obscurity of the text.

5.2.4.5 Error Handling: Languages and scripts other than English and Alphabet will be ignored so as to not generate comments that are not applicable for the text.

**5.3 Non-functional Requirements**

5.3.1 Usability

The product will be a stand-alone application that will run on windows platform. It can be used on any machine that has Windows OS version 7 onwards. It will be useful for students, writers or anybody who needs to write a document in a particular style.

5.3.2 Reliability

The software will work only for English language and is not meant to be used for any other language. For certain functionalities, the usage is to be limited to American English only.

5.3.3 Implementation

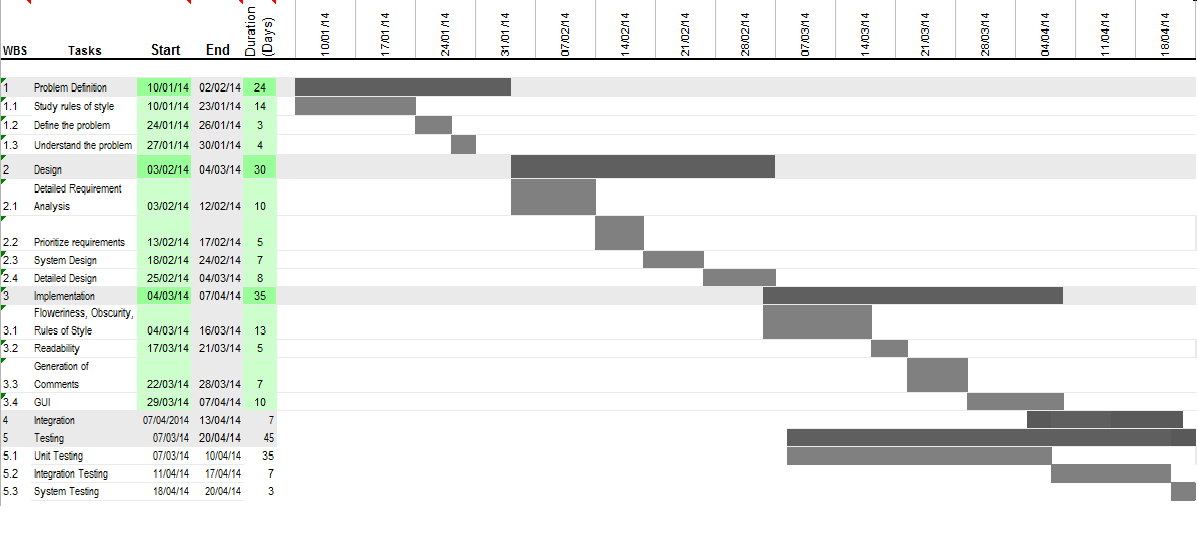
The implementation of the software will be in Python language. NLTK framework will be used for all language processing work and NLTK Data comprising of corpora such as Brown, Wordnet and dictionaries such as CMU will be used for referencing purposes.

5.3.4 Interface

The software will have a GUI for the text editor to facilitate input of various parameters (Target readability, writing style, etc), text to be processed and the comments and suggestions generated by the software.

**CHAPTER 6**

**GANTT CHART**

****

***Fig. 6.1*** *Gantt chart*

**CHAPTER 7**

**SYSTEM DESIGN**

**7.1 Block Diagram**

The system is divided into the following subsystems:

****

***Fig 7.1*** *Block diagram of the system*

7.1.1 Python Back-end

The software will analyse the text input by the user and analyze it for different

rules and styles. This will be in turn be using NLTK framework and the data

available in various corpora and dictionaries, etc.

7.1.2 User Subsystem

The user subsystem will represent the users of this system. The user will interact with our system by creating/opening a .txt file in the editor window.

7.1.3 NLTK

All language processing will be done using NLTK framework. It is an open

source, free toolkit that is written in python.

7.1.4 NLTK data

The NLTK framework comes with support for various corpora, dictionaries, etc.

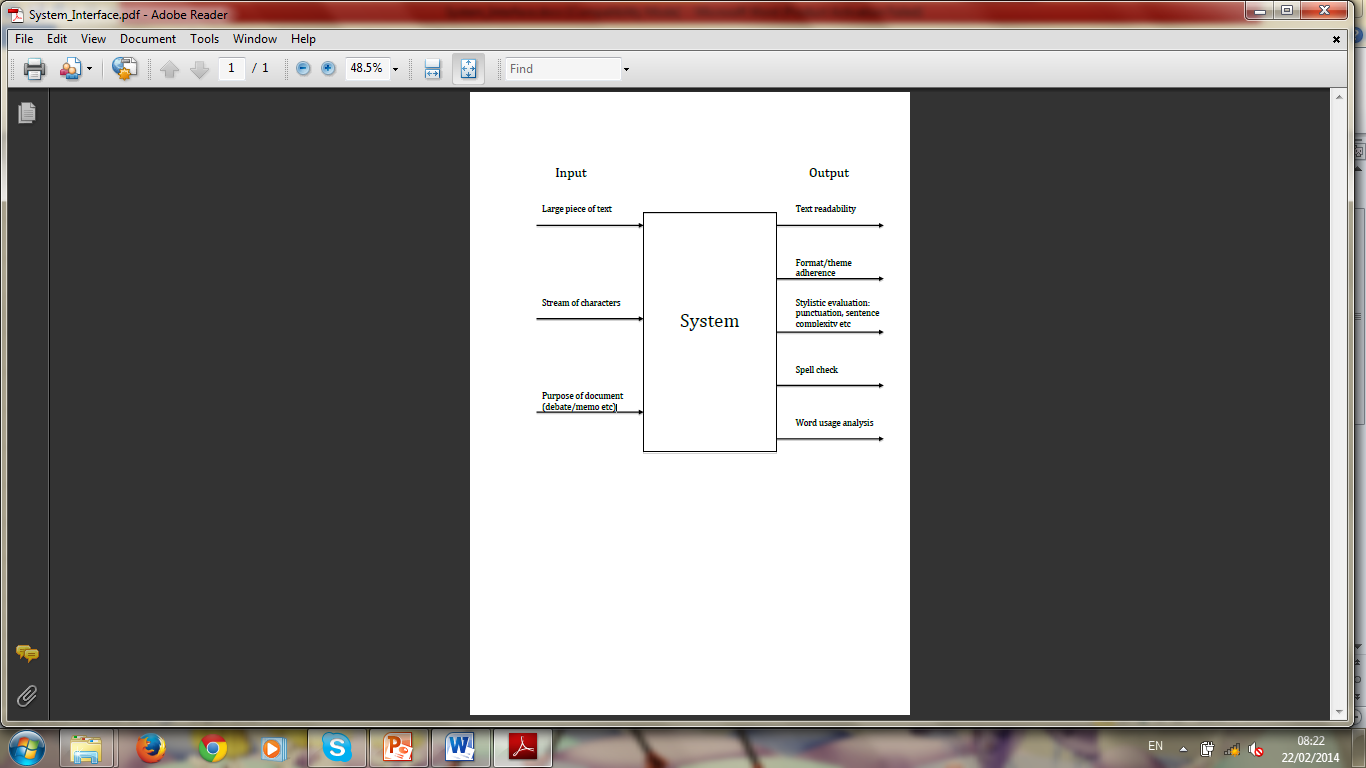
The corpora used are Brown and Wordnet. CMU dictionary is also used as a reference for some functionality.

7.1.5 PyQT User Interface

The user interacts with the system by the means of an editor window. This subsystem is responsible for the flow of data to the back end and display of comments, etc.

**7.2 Architecture**

As mentioned earlier the user interacts with the system through an editor window. The user can choose to either type directly into the window or open an existing file. The user can also choose to specify the purpose of the document. The output generated is the readability of the text, template for the chosen format, stylistic evaluation, word usage analysis, floweriness, obscurity, etc.



***Fig 7.2*** *Architecture for the proposed system*

The major phases in the processing of the text input by the user are as follows:

* Creating and saving a text file
* Giving the user a template for the formats
* Sentence level analysis
* Word level analysis
* Calculating readability of the text
* Calculating floweriness and obscurity of the text
* Generating comments in the editor window

7.2.1 Creating and saving a text file

The text to be processed can be directly typed into the editor window, saved into a new file or loaded from an existing file.

//sidhi – add details

7.2.2 Giving the user a template for the formats

The following templates are provided to the user for consideration. These templates open a dialogue box when chosen. There are various text boxes in the dialogue box, one for each section of that particular format.

7.2.2.1 Formal Letter: A formal letter should have the following:

* Sender’s address
* Receiver’s Name
* Receiver’s Address
* Date
* Subject
* Salutation
* Content
* Regards
* Sender’s Name

7.2.2.1 Debate: A debate should have the following:

* Introduction
* Argument 1 followed by justification
* Argument 2 followed by justification
* Argument 3 followed by justification
* Conslusion

7.2.2.3 School Essay: A good school essay should have the following:

* Title of the essay
* Introduction of the topic
* Main point #1 followed by its description
* Main point #1 followed by its description
* Main point #1 followed by its description
* Conclusion

7.2.3 Sentence level processing

The following is done as a part of sentence level analysis of the text:

7.2.3.1 Sentence tokenization: The text is broken into sentences for them to the analysed. This is done using the functions provided by NLTK.

7.2.3.2 Check if sentence is too long: The length of the sentence is compared with a hard-coded value that generates comments about the sentence being too long if true.

7.2.3.3 Check if the sentence is in passive voice. If the use of passive voice is detected in a sentence, appropriate comments are generated in the editor window.

7.2.3.4 Check if any commonly misused phrases are present in the sentence: A hard-coded list of misused phrases is looked for in the text. If any of them are present, corrections are suggested.

7.2.4 Word level processing

The word level processing of the text involves the following:

7.2.4.1 Tokenizing each sentence into words: Word tokenizing is done using functions provided by the NLTK.

7.2.4.2 Tagging of words: The tokens (words) are tagged as parts of speech based on their context in the text. The POS tagging feature of NLTK is used for this purpose.

7.2.4.3 Check if the last word in the sentence is a preposition: Based on the tag of the last word in the sentence, it is judged whether the sentence ends with a preposition or not

7.2.4.4 Check if the words are misused: The tagged words are compared against some hardcoded values. If the tag doesn’t match the hard-coded value, alternate words are suggested.

7.2.5 Calculating readability of the text: The following reading scores are generated:

7.2.5.1: Flesch Reading Ease:

206.835 - 1.015 \* (total words / total sentences) - 84.6 \* (total syllables / total words)

The higher the score, the easier it is to read.

7.2.5.2 Flesch-Kincaid Grade Level:

0.39 (total words / total sentences) + 11.8 (total syllables / total words) - 15.59

Lowest theoretical grade score: -3.4

7.2.5.3 Gunning-Fog Score:

0.4 \* [ (words / sentences) + 100 \* (complex words / words) ]

7.2.5.4 Coleman-Liau Index:

0.0588L - 0.296S - 15.8

L: average number of letters per 100 words

S: average number of sentences per 100 words.

7.2.5.5 SMOG Index:

1.043 \* √ (number of polysyllables \* 30 / number of sentences) + 3.1291

polysyllables: words with 3 or more syllables

7.2.5.6 Automated Readability Index:

4.71 \* (characters / words) + 0.5 \* (words / sentences) - 21.43

7.2.6 Calculating floweriness and obscurity of the text

7.2.6.1 Floweriness: The floweriness of the text is determined by the number of adjectives and adverbs per sentence in the text. The tag of each word in the sentence is checked. All the words tagged adjectives and adverbs are counted. This number is divided by the total number of sentences in order to generate the floweriness

7.2.6.2 Obscurity: The frequency (its usage in English language) of each word in the sentence is found. The words with frequency less than 5 are considered rare/difficult. All such words are counted and the total divided by the total number of sentences in the text to generate the obscurity.

7.2.7 Displaying comments in the editor window:

All the comments generated by the back-end subsystem need to be displayed in the main editor window. The readability scores are calculated either automatically (triggered by the user by pressing the space bar) or manually, when the user presses either Ctrl+G or clicks on the calculate readability score under the tools section.

The evaluation of the text is done when the user presses Ctrl+E or clicks on Evaluate text option under the tools menu.

**7.3 Use cases:**

3.4.1 Creating a document

3.4.1.1 Actor: Generic user

3.4.1.2 Description: The user can use the application to create a document.

Once the file has been created, he/she can type text and save the document.

3.4.1.3 Pre-condition: Unopened application.

3.4.1.4 Post-condition: Blank canvas for the user to start typing text.

3.4.2 Evaluating stylistically

3.4.2.1 Actor: Generic user

3.4.2.2 Description: The user can use the application to obtain comments on his style of writing. The application detects passive voice usage, inconsistent tenses, provides information on word count, etc.

3.4.2.3 Pre-condition: A substantial amount of text written by the user.

3.4.2.4 Post-condition: Comments against the corresponding line number.

3.4.3 Checking word usage and word frequency

3.4.3.1 Actor: Generic user

3.4.3.2 Description: The user can use the application to obtain data on how common the words he has written in the document are and words that he has used too often.

3.4.3.3 Pre-condition: A substantial amount of text written by the user.

3.4.3.4 Post-condition: Comments on words that have been used very frequently. A rating of the complexity of the text based on universal word frequency.

3.4.4 Conforming to a template

3.4.4.1 Actor: Student

3.4.4.2 Description: A student can use the application to adhere to a particular template while writing a document. This is a preference given prior to document editing and hints are provided as the student prepares the document.

3.4.4.3 Pre-condition A blank document

3.4.4.4 Post-condition Hints to assist the student to adhere to the template (ex: debate)

3.4.5 Analysing readability

3.4.5.1 Actor: Novelist

3.4.5.2 Description: The novelist or writer can use the application to obtain an analysis on the readability of the text written. The readability scores are calculated from universally accepted formulae.

3.4.5.3 Pre-condition: A complete document

3.4.5.4 Post-condition: Readability scores on the document.

**CHAPTER 8**

**DETAILED DESIGN**

**.1 Application architecture**

The Input 🡪 Process 🡪 Output model is followed for each of the following stages:

* Stylistic evaluation and generation of comments
* Display of comments
* Text highlighting
* Adherence to templates
* Conformity to target readability
* Opening an existing file
* Saving an edited file
  + 1. Stylistic evaluation and generation of comments

Input Process Output

**Text for evaluation**

**Evaluate against rules of style**

**Comments**

***Fig 8.1.1*** *Generation of comments*

The text is taken for evaluation against a predefined set of rules of style. Comments are generated for every violation of rules of style. Each comment has a priority depending on the importance of that rule of style.

* + 1. Display of comments

Input Process Output

**Comments**

**Display of comments as in list in GUI**

**Colour-coded comment list**

***Fig 8.1.2*** *Display of comments*

The comments generated after evaluation of the text should be displayed to the user. They appear as a list. The colour of each comment depends on the priority assigned to it. The user can view comments of only one priority or all comments.

* + 1. Text highlighting

Input Process Output

**Clicked comment**

**Text search**

**Highlighting of text**

***Fig 8.1.3*** *Text highlighting*

When the user clicks on a comment, the text to which the comment refers gets highlighted. Even if the text is not visible on the screen, the file scrolls down so that the text is visible when it is highlighted.

* + 1. Adherence to templates

Input Process Output

**Template option**

**Displaying the template for user to insert text**

**Text in editor window**

***Fig 8.1.4*** *Adherence to templates*

When the user chooses a template to follow, the template appears along with boxes to the user to enter textual data. Upon completion of this action, the text along with help text appears in the editor window. The user can further edit this.

* + 1. Conformity to target readability

Input Process Output

**Target readability**

**Evaluation of readability**

**Readability highlighting**

***Fig. 8.1.5*** *Conformity to target readability*

The user selects a target readability score. The next time the user requests for readability analysis, the results are highlighted if the text does not lie in the target readability range. In case of auto evaluation of readability, the results are highlighted as and when the user types in the editor window.

* + 1. Opening an existing file

Input Process Output

**File name**

**Opening file**

**File contents in editor window**

***Fig.8.1.6*** *Opening an existing file*

If the user wishes to open an existing file he may search in the system directory and chose a file. That file is read and its contents are visible in the editor window.

* + 1. Saving an edited file

Input Process Output

**File with text**

**Transferring contents to a file**

**Text**

***Fig 8.1.7*** *Saving an edited file*

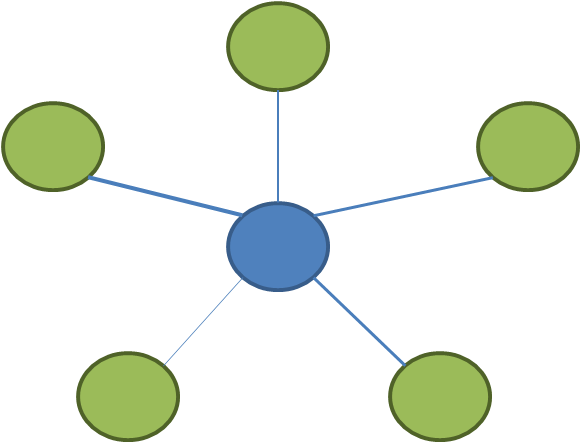
When the user wishes to save his work, if the file already exists, then it’s contents are replaced by the current text in editor window. However, if the file does not exist, then the user must choose a file name and the directory in which he wishes to save the file. The file data can now be accessed outside the application.

**8.2 Modules**

8.2.1 Generation of ego networks

Once we have the user profiles of all the users in the required UDF, the ego networks are generated for all the users by following the below mentioned steps:

* All the attributes that are static (or will not change) and all the ones that are dynamic (change often), are identified.
* Attribute matching is done for all the users in the network, with the attribute of the user, and two levels of matching can be identified. The first level indicates that the matching is exact and the second level indicates that the matching is similar but not exact.
* The ego attribute values that come under first level are given a higher preference to the ones that come under second level.
* Based on this matching, the users who match exactly are identified and the users who match similarly are identified. From this, a star network can be created for the user under consideration. This will give us the ego network for the user who is under consideration.

****

***Fig.8.2.1*** *Ego Network*

8.2.2 Context creation

Context creation is done with the help of recommendation context creation is expected to be done by context users. This is done in 2 steps:

* Naming
* Properties identification

8.2.2.1 Naming

* User enters the description of the contexts. Using this description, a term is found for that description by making a call to the reverse dictionary.
* The context description is saved in the context tree/database.
* The reverse dictionary returns n terms (usually 10) that will match the description given by the user.
* If the user agrees with the results retrieved, he uses one of the terms as the context name.
* If the user doesn’t agree with the results retrieved, he may enter his own name or the user will repeat the procedure with some different description. In such a case, the context database will be updated.

8.2.2.2 Properties identification

Context is some situation or circumstance which is useful for the user. To make it useful, right users have to be found. In order to do that, the context is expressed in terms of properties. There are two types of properties.

* Quantitative Properties
* Qualitative Properties

Quantitative Properties are directly measurable.

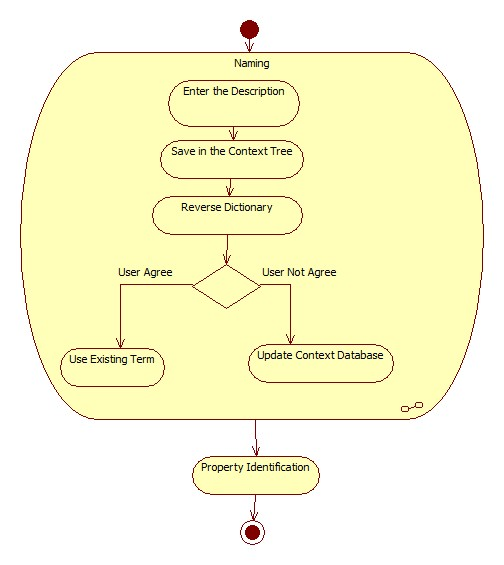
* The system will recommend the available attributes.
* User chooses one of the attribute as the property name.
* Use enters values. These values can be:
  + Tuple, List
  + Range (Numerical)
  + Single value
* Attribute, along with values, is stored in the context database.
* The same process is repeated for the required number of times.

Qualitative Properties are not directly measurable.

* User enters the description of the property.
* Using the reverse dictionary look up, the property names are suggested to the user.
* The property description is saved in the context database.
* If the user accepts any of the suggested names, it is stored in the database.
* If the user doesn’t agree with the results, the user will provide a name, or repeat the same procedure with different description, and the context database gets updated.

The following assumptions are made:

* Reverse dictionary look up can be made.
* Context database is kept updated.
* Description of the context name, along with description of attributes, is important in order to generate congruent networks.



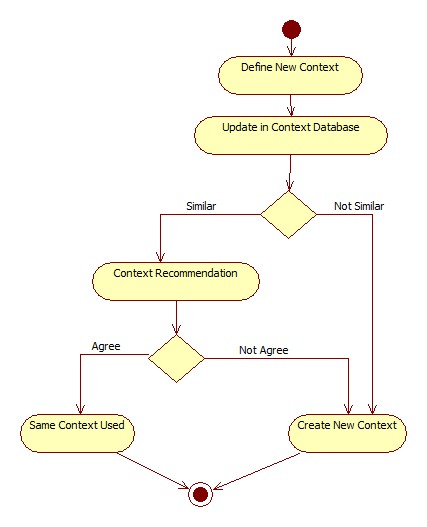
***Fig. 8.2.2*** *Activity diagram for creating of new contexts*

8.2.3 Use of existing contexts

* When a user wants to define a new context, he enters description of the context.
* This description is compared with the description of the existing contexts in the database.
* If the descriptions are found to be similar, these contexts will be recommended to the user, so that the user can select the existing context.
* Recommendation also includes properties of the context, along with their description.
* If the agrees, the same context is used. A congruent network would have already been generated, and the same congruent network will be used with some updates.

The following assumptions are made:

* Facility exists to compare a context and find similar contexts.
* Facility exists to update congruent networks.



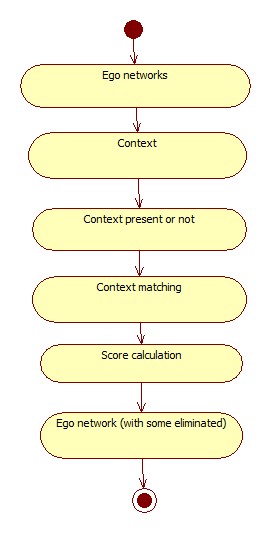
***Fig. 8.2.3*** *Activity diagram for using an existing context*

8.2.4 Elimination of ego networks

* For the given context, identify the attributes that describes that context best
* Context matching is done between the attributes of the context and the attributes of the ego node in the egocentric network
* If CA ∩ NA < Threshold, then eliminate the ego network

CA

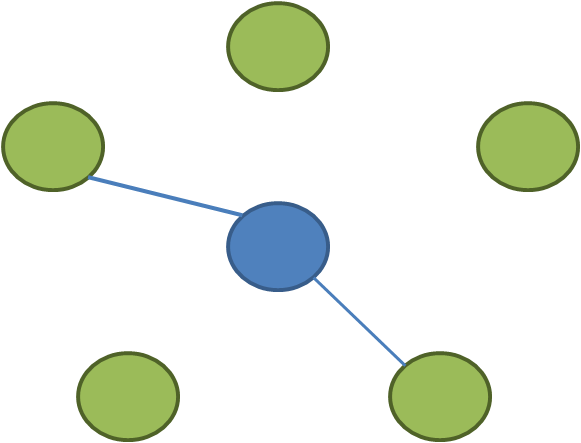
* Only the ego networks whose score is above this threshold value are considered for the next steps.



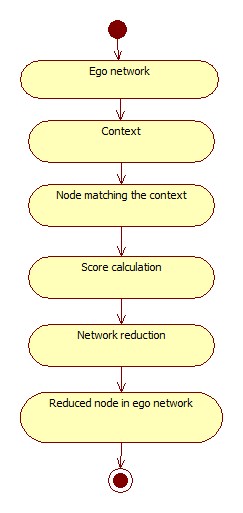
***Fig. 8.2.4*** *Activity diagram for elimination of Ego networks*

8.2.5 Reducing the number of nodes in each of the remaining egocentric network

* For each of the remaining egocentric network
* Nodes whose score is less than the threshold are eliminated and we are left with the ego centric network for the particular context.
* Hence, we derive the context specific egocentric network
* At the end of this step we would be left with certain number of context specific egocentric network



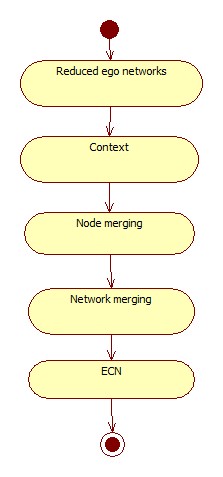
***Fig.8.2.5.1*** *Ego network with nodes reduced*



***Fig.8.2.5.2*** *Activity diagram for reduction of nodes in ego network*

8.2.6 Generation of ECN

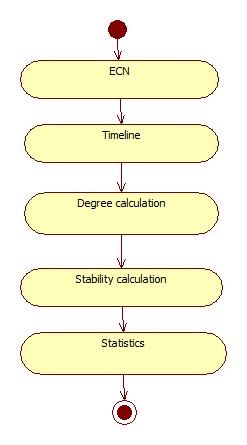
* We consider all the contest specific egocentric networks for this step.
* A network of all the nodes, in the remaining networks, is formed to generate the egocentric congruent network.



***Fig.8.2.6*** *Activity diagram for generation of ECN*

8.2.7 Determine the degree and stability of ECN

* Degree tells us to what extent the nodes of the ECN are congruent with each other
* Stability of the ECN tells us the number of nodes that “join” or “leave” the congruent network, is the specified timeline
* To simulate this, the user profile information is updated, and based on this modification, the ECN regenerated for the same context. The number of nodes that “come into “ECN or “go out” of the ECN, tell us about the stability
* Higher this number, lower is the stability of the ECN

****

***Fig. 8.2.7*** *Activity diagram for measurement of degree and stability of ECN*

**8.3 Database Design**

The number and type of comments are fixed in the application. Another candidate for usage of databases is misused words/phrases. But the number of misused words/phrases is fixed and more will not be added during program execution. Hence there is no reason to bring in databases.

However, data must still be stored. Misused words/phrases are stored in the form of a list of tuples.

Misused words:

[(‘word1’, ‘tag1’, ‘replacementWord1’), (‘word2’, ‘tag2’, ‘replacementWord2’), …,]

Misused phrases:

[(‘misusedPhrase1’, ‘correctPhrase1’), (‘misusedPhrase1’, ‘correctPhrase1’), …,]

**CHAPTER 9**

**IMPLEMENTATION**

**9.1 Implementation choices**

The options for the programming language for implementation mainly included C++, Java, C# and Python.

The system needed a natural language processing subsystem at the back end. The options available were Apache Open NLP, Sharp NLP and NLTK.

The considered choices for implementing the front end were Camelot, PySide, PyObjC, QT, PyQT and wxPython.

**9.2 Reasons for choosing a particular implementation**

Python was chosen as the language for implementing the back end due to its support of various NLP libraries and UI frameworks. Also, application programming using python was considered more suitable. The Python regex library is also used in the implementation, and was found very useful.

NLTK, or Natural Language Toolkit was chosen for this purpose due to its easy-to-use interfaces with many corpora and lexical resources such as wordnet, CMU dictionary, Brown Corpus, etc. NLTK also comes with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.  NLTK is a free, open source, community-driven project available for Windows, Mac OS X, and Linux. NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike.

The front end, i.e. the UI was created using PyQT version 4.10.3PyQT allows creation of the GUI with the help of QT designer that is easy to use. The .ui file can then directly be converted to .py files by using pyuic.

**9.3 Data structures used**

The data structures used to store the following details are:

* Misused words: A list with each element being a list itself. Each element has three strings – the misused word, it’s tag, correct word
* Misused phrases in Indian English: A list with each element being a list of two strings: the first one is the incorrect phrase, second is the suggested correction.
* Comments generated: The comments are stored in a list with each element containing the three things:
  + The sentence for which the comment was generated as a string
  + The severity of the error found
  + The Comment as a string
* Floweriness:
* Tokenised sentences: The nltk.sent\_tokenize() function returns a list of tokenised sentences.
* Tokenised words: The nltk.word\_tokenize() function returns a list of tokenised words.
* Tagged words: The nltk.pos\_tag () function returns a list of tagged tuples.
* Readability:
* Statistics about the text: All the statistics about the text such as number of sentences, number of words, number of syllables, number of polysyllables and number of characters are stored in a list.
* Syllable counting: A list is used for finding the number of syllables in the text, further used to calculate the readability.

**9.4 Communication mechanisms used and their structure**

* QT Action

The QAction class provides an abstract user interface action that can be inserted into widgets. In applications many common commands can be invoked via menus, toolbar buttons, and keyboard shortcuts. Since the user expects each command to be performed in the same way, regardless of the user interface used, it is useful to represent each command as an *action*.

* QT Signal/Slot mechanism

The signal/slot mechanism has the following features:

* + A signal may be connected to many slots.
  + A signal may also be connected to another signal.
  + Signal arguments may be any Python type.
  + A slot may be connected to many signals.
  + Connections may be direct (ie. synchronous) or queued (ie. asynchronous).
  + Connections may be made across threads.
  + Signals may be disconnected.

**9.5 Psuedocode**

1. For detection and correction of misused phrases in Indian English:

For each sentence in the input data

For each element in the list of misused phrases:

If the first string exists in the sentence,

Generate a comment for the sentence.

2. To calculate obscurity:

Load the word frequency distribution from Brown Corpus.

For each word, get the frequency of the word from Brown Corpus.

If the frequency is less than 5, increment count.

Divide the total count by the number of sentences to get obscurity.

3. To calculate floweriness:

Initialise adjCount as 0.

For each sentence, tag the words in the sentence.

Initialise adj, adv to 0.

For each tagged word, check if the tag is JJ or JJR or JJS. If yes,

Increment adj

For each tagged word, check if the tag is RB or RBR or RBS. If yes,

Increment adv

Add adj and adv to the adjCount

4. Checking if a sentence ends with a preposition:

For each sentence:

Get the number of words in the sentence

Compare the tag of the last word with IN. If yes,

Return true.

5. Detecting tense inconsistency:

For each sentence:

Set isPast, isPresent and quotes to false

For each tagged tuple in the sentence:

If the first element is single or double quote,

Set quotes to true

If quotes is false:

If tag is Past tense verb:

If isPresent is true,

Return false

Else:

Set isPast to true

Else if tag is present tense verb:

If isPast is true,

Return false

Else:

Set isPresent to true.

5. Checking length of the sentence:

For each sentence:

Tokenize each sentence into words.

Count the number of tokens.

If length is more than 25, generate comment about sentence being too long.

6. Detecting misused words:

For each sentence, search if any misused word is present in the sentence.

If yes:

Check if the tag matches what it should when used in the right context.. If not:

Generate comment suggesting the correct word.

7. Calculating Readability

Flesch Reading Ease:

206.835 - 1.015 \* (total words / total sentences) - 84.6 \* (total syllables / total words)

The higher the score, the easier it is to read.

Score range Implication

90-100 11 year old student

60-70 13-15 year old students

0-30 university graduate students

Readability formulae:

Give an approximation of the readability of the text by specifying the years of formal education needed to understand the text on a first reading. Hence, the higher the score, the tougher the text is to read.

Each have their own formula considering various parameters.

Flesch-Kincaid Grade Level:

0.39 (total words / total sentences) + 11.8 (total syllables / total words) - 15.59

Lowest theoretical grade score: -3.4

Gunning-Fog Score:

0.4 \* [ (words / sentences) + 100 \* (complex words / words) ]

Complex words:

- words with 3 or more syllables.

- proper nouns not included

- common suffixes like -es, -ed, or -ing are excluded

Score of 8 for near universal understanding.

Score of <12 for a wide general audience.

Coleman-Liau Index:

0.0588L - 0.296S - 15.8

L: average number of letters per 100 words

S: average number of sentences per 100 words.

SMOG Index:

1.043 \* √ (number of polysyllables \* 30 / number of sentences) + 3.1291

polysyllables: words with 3 or more syllables

Sample size of text must be at least 30 sentences.

Automated Readability Index:

4.71 \* (characters / words) + 0.5 \* (words / sentences) - 21.43

**CHAPTER 10**

**INTEGRATION**

**10.1 Integration strategies**

The development part of this project follows Bottom up approach. The bottom up approach is used for building small components, and finally to integrate them into one complete system. Initially the backend implementation was done. Each small component was implemented in python. API's were defined for better interaction and better integration with the frontend as well as for interaction between the backend modules.

To integrate all the components, top down integration approach is used. Top down approach is better for integration, and for every scenario, the corresponding backend API to be used, is also known. The User interface functionalities are linked with backend one after the other one, in a sequential manner. For the frontend, HTML has been used and the backend coding has been done using Python. As python was used, integration was a challenging task. No python frameworks such as Django were used. Python is implemented as a web service by using CGI scripting. After that, the User Interface and the backend CGI scripting was integrated in a phase wise or stage wise manner.

**10.2 User Data Generation**

Using Graph API, the Facebook Access token is obtained. Using this access token, the user profile information and his connected users’ information is fetched in a JSON format. From this JSON data, the required field values is queried and stored in database as User Profile(UP) format.

Users data in UP format

Fetch the User’s Information from facebook

Get Facebook Access token

***Fig. 10.1*** *User data generation*

**10.3 Pools Creation**

From the user profile stored in the database, each user’s information is fetched. This information is parsed and user attributes are retrieved for the different pools. The pools are created by the attribute names of the user fetched from the database. These pools are filled with attribute values. This is done for each of the users’ information and the various pools are generated.

User profile

Get Users Attributes

Parse the Attributes

pool

***Fig 10.2*** *Pool creation*

**10.4 Description Builder**

The pools information present in database is fetched. Each pool is parsed and the attribute values are taken, the description for the attribute value is fetched from the Wikipedia, WolframAlpha and Wordnet.

Get Attributes from Pools

Description for the Attributes

pool

Description builder

***Fig.10.3*** *Description builder*

**10.5 Similarity calculation**

Similarities are calculated between the context entered by the user and the contexts in the database. The user enters the description of the context and the attributes for that context. This context will be matched with the existing contexts in the context database, and the best matching contexts are recommended to the user.

**10.6 Add Context**

In this User Interface component, the Context description is taken as input from the user. First, contexts are recommended from the existing contexts. If the user does not select the context from it, using the Reverse Dictionary the possible context names are suggested. User can select the suggested context name or user can enter the new context name for the description, and the context name and context description are stored in the database. After this, the user is asked to enter the Property description, similar as Context description the property name is suggested to user. Here too, the user can select the suggested property name or he can enter new property name and update the same in the database for that context.

Suggest Names

Get Context Description

Ask for Property Name and Description

Select the Context Name for the Description

Context database

***Fig. 10.4*** *Add Context*

**10.7 Browse Context**

In this User Interface component, the context names present in database are fetched from the database and displayed. User can click any of the context name displayed, to see the context description. The required context name from the context list is to be entered to choose the context.

Suggest Context Names

Select the Context

Context

Database

***Fig.10.5*** *Browse context*

**10.8 Generate Network**

Context description, properties and property description for the corresponding context are fetched. The fetched property description is compared with the pools description. The pool description which matches the properties is fetched from database. These selected pools property values are compared with the user profile. The user profiles which match the property values are selected. The selected user IDs and their names are used to generate the graph. Nodes in the network graph generated represent users congruent to the context. Graph generated contains users who match to the context, containing connections between them. User can click any of the nodes present in graph. The user information of the clicked node is fetched from the database and displayed.

Users’ id to Graph UI

Match (Property Value, User Property Value)

Fetch the Properties for the given Context

Match (Property Value with User Property Value)

Context

database

User Profile

***Fig. 10.6*** *Generate Network*

User Data Collection

User Profile

Pools Creation

Pools

Description Builder

Add Context

Context database

Browse Context

Generate Graph

***Fig. 10.7*** *Integrated System*

The user data collected from Facebook is inserted into MongoDB in required format. From this data, pools are generated. Description of every attribute value in all the pools is generated from Wikipedia, Wordnet, and WolframAlpha and stored. For every pool, similarity between attribute values is calculated and stored.

When a new context is added by a user, the context database is updated accordingly. When a user wants to use an existing context, he can browse and select the context. Once the user has entered a context, his Egocentric congruent network is generated by considering the user’s profile information and the context in which he is in. This network is generated by matching the user’s profile with other users who are also in that context. The Integrated system which generates the network is as shown in the diagram above.

**CHAPTER 11**

**TESTING**

**11.1 Unit Testing**

11.1.1 File options

* *Test Definition:* Checking file options of new, open, save, close
* *Entry criteria:* The application must be open
* *Exit Criteria:* The application is closed.
* *Test input:* Name of the document
* *Test Output:* New document is created/existing document is opened, modified, saved and closed

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Expected Result** | **Actual Result** | **Conclusion** |
| If the currently open file is modified “save?” option must appear | User must be given the option of saving current file | User is given the option of saving current file | Success |
| Create a blank new document | Blank new document is created and opened | Blank new document is created and opened | Success |
| Open an existing file | Existing file opens. | Existing file is opened | Success |
| Open a non-existent file | Failure in opening file | File opening failed | Success |
| Upon opening an existing file, display of file name in title bar | Display of file name is title bar | File name is displayed in title bar | Success |
| Saving a file after it is modified | Give save option | Save option given | Success |
| Closing an unsaved file | Give save option | Save option given | Success |

11.1.2 File’s edit options

* *Test Definition:* Editing a file*.*
* *Entry Criteria:* The application is open*.*
* *Exit Criteria:* All edit options are tested
* *Test Input:* Text
* *Test Output:* Text

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Expected Result** | **Actual Result** | **Conclusion** |
| Copying/cutting and pasting after selecting text | Write same text again | Text is copied/cut and pasted | Success |
| Copying text when nothing is selected | Disable copy option | Copy option disabled | Success |
| Undo when no changes are made in the editor | Disable undo option | Undo option disabled | Success |
| Redo when there is no prior undo | Disable redo option | Redo option disabled | Success |
| Redo changes undone | Perform redo | Redo is performed | Success |

11.1.3 Formats

* *Test Definition:* Typing text in a format*.*
* *Entry Criteria:* A format UI is open*.*
* *Exit Criteria:* Format data appears in editor window
* *Requirement:* “OK” is pressed in the format window
* *Test Input:* Text
* *Test Output:* Text in editor window

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Expected Result** | **Actual Result** | **Conclusion** |
| Filling all text boxes in the format | Write text in the order entered, into the editor window | The text appears in the order entered, into the editor window | Success |
| Leaving a few text boxes blank in the format | Text written, along with help text for those fields left blank | Text written, along with help text for those fields left blank appears in editor window | Success |
| All fields left blank | Write help text for all fields in to editor window | Help text for all fields appears in editor window | Success |

11.1.4 Stylistic comments

* *Test Definition:* Stylistic comment list
* *Entry Criteria:* A list of stylistic comments
* *Exit Criteria:* Highlighting text and displaying comments only of the colour specified
* *Requirement:* Comments of all colours and multiple comments for the same sentence
* *Test Input:* Text and colour option
* *Test Output:* Text highlighting

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Expected Result** | **Actual Result** | **Conclusion** |
| Clicking a comment | Highlighting the sentence associated with the comment | The sentence associated with the comment is highlighted | Success |
| Changing the text and clicking a related comment | No highlighting | No text is highlighted | Success |
| Choosing a comment colour option | Displaying comments only of that colour | Comments only of that colour are displayed | Success |

11.1.5 Stylistic analysis

* *Test Definition:* Detection of stylistic errors
* *Entry Criteria:* A piece of text
* *Exit Criteria:* Completion of stylistic analysis
* *Requirement:* Grammatically correct text
* *Test Input:* Text data
* *Test Output:* Presence or absence of stylistic error

11.1.5.1 Test cases for rules of style

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Actual Result** | **Conclusion** |
| Detection of passive voice | A sentence with simple passive voice | Passive voice detected | Success |
|  | A sentence in active voice | Passive voice not detected | Success |
| Detection of a preposition ending | A sentence with a preposition at the end | Preposition ending detected | Success |
|  | A sentence without a preposition at the end | Preposition ending not detected | Success |
| Detection of multiple tenses in the same sentence | A sentence with multiple tenses | Tense inconsistency detected | Success |
|  | A sentence in a single tense | Tense inconsistency not detected | Success |
| Detection of misuse of words | A sentence with “then” instead of “than” | Misused word detected | Success |
| Detection of Indian English | A sentence with “yesterday night” instead of “last night” | Indian English detected | Success |
| Detection of long sentences | A sentence with length > 25 | Long sentence detected | Success |

11.1.5.2 Test cases for stylistic statistics

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Actual Result** | **Conclusion** |
| Floweriness | A sentence with 2 adjectives and 3 adverbs | Floweriness is 5 | Success |
| Obscurity | A sentence with 2 infrequent or rare words of English | Obscurity is 2 | Success |

**11.2 Integration testing**

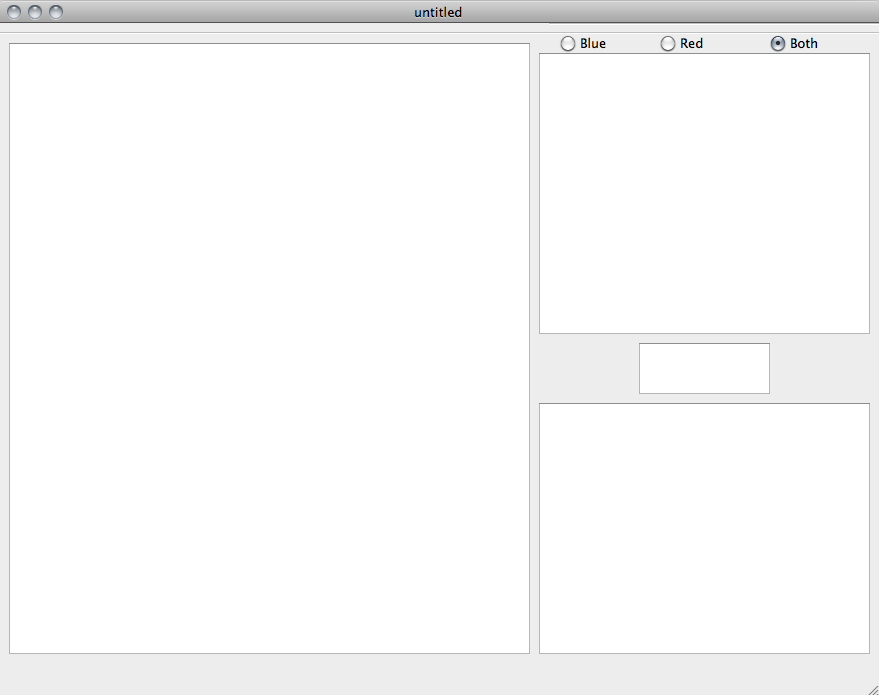
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Component** | **Number of Test Cases** | **Test case Description** | **Result** | **Conclusion** | **Issues faced and fixed/ Comments** |
| Opening text file | 4 | Create a new file | Text file created | Success |  |
|  |  | Open an existing file | Text from existing file loaded into the editor window | Success | No comments |
|  |  | Try to open an existing file and then cancel | Existing file not opened | Success |  |
|  |  | Type into an untitled file and then save it | File saved in the target directory | Success |  |
| Evaluating Readability | 4 | Evaluate readability of a blank document by clicking on Evaluate readability | Readability is 0 | Success | No Comments |
|  |  | Evaluate readability of the text while typing | Appropriate readability displayed | Success | No comments |
|  |  | Close the document and reload it. Check readability now | Readability remains the same as the previous time | Success | No Comments |
|  |  | Check readability while typing by enabling auto-evaluate | Readability evaluated every time there is the space bar is pressed | Success | Event is triggered when the space is encountered |
| Setting Target Readability | 3 | Set readability as Children  Test a document with a higher readability | Readability comments change colour | Success | No comments |
|  |  | Test an existing document for different readabilities. Do not click Evaluate | No change | Success | No comments |
|  |  | Test an existing document with different readabilities by pressing evaluate every time. | Colour changes for everything except target readability | Success | No Comments |
| Generating comments | 11 | Give a poem as a text | Proper comments not generated | Failed | Limitation of the parsing system being used, in this case NLTK |
|  |  | Use multiple tenses in the same sentence | Appropriate comments generated | Success | The writer should stick to one tense in a sentence |
|  |  | Multiple tenses across sentences in a paragraph | No Comments generated | Success | Checking of tenses is a sentence level functionality |
|  |  | Sentence with 30 words | “Sentence may be too long” comment generated | Success | The sentence should not be more than 25 words. |
|  |  | Sentence with 20 words | No comment generated | Success | No comments |
|  |  | Sentence with 3 adjectives in a sentence | Floweriness: 3.0 per sentence | Success | Floweriness = Number of Adjectives per sentence |
|  |  | Sentence with no adjectives | Floweriness  : 0.0 per sentence | Success | Floweriness = Number of Adjectives per sentence |
|  |  | Sentence with passive voice | Passive Voice comment generated | Success |  |
|  |  | Sentence in active voice | Passive voice comment not generated | Success |  |
|  |  | Sentence with 3 rare words in a sentence | Comment: Obscurity: 3.0 per sentence | Success | Obscurity = Number of words with frequency <5 per sentence |
|  |  | Sentence with 0 rare words in a sentence | Comment: Obscurity: 0.0 per sentence | Success | Obscurity = Number of words with frequency <5 per sentence |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**CHAPTER 12**

**STORY BOARD**

**12.1 Blank document page**

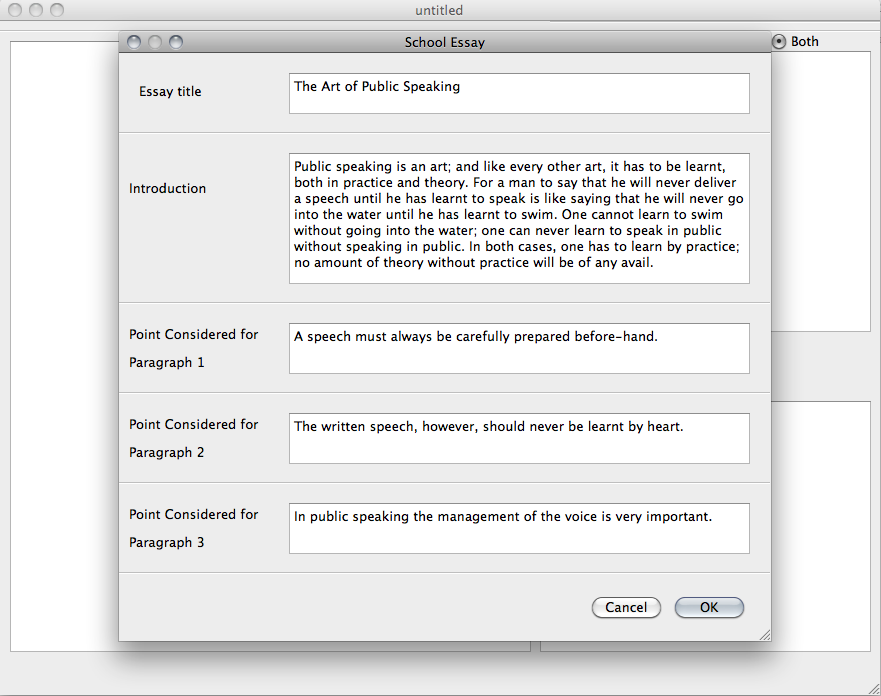
When the user first opens the application, a blank editor appears. From here, the user can chose to type in the blank document and hence create a blank file or to open an existing file. If an existing file is opened the contents of the file appear in the editor section.



***Fig. 12.1*** *Blank doc page*

**12.2 Writing in a template**

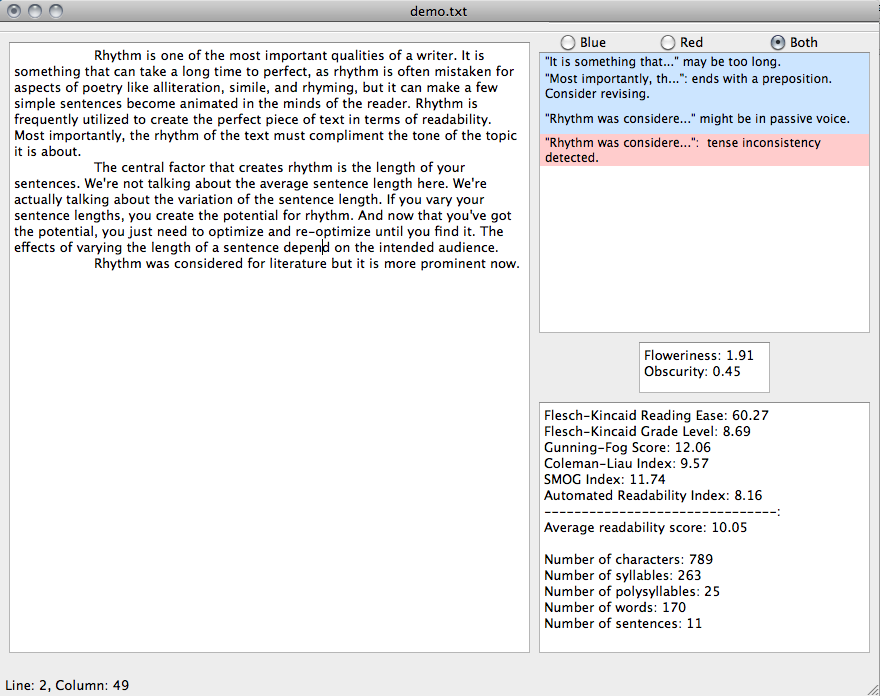
The user can chose from several templates. Here, a sample of school essay is considered. After the fields are filled in, the contents appear on the editor section. Along with the fields, additional help text also appear.



***Fig.******12.2*** *School essay template*

**12.3 Evaluating for stylistics and readability**

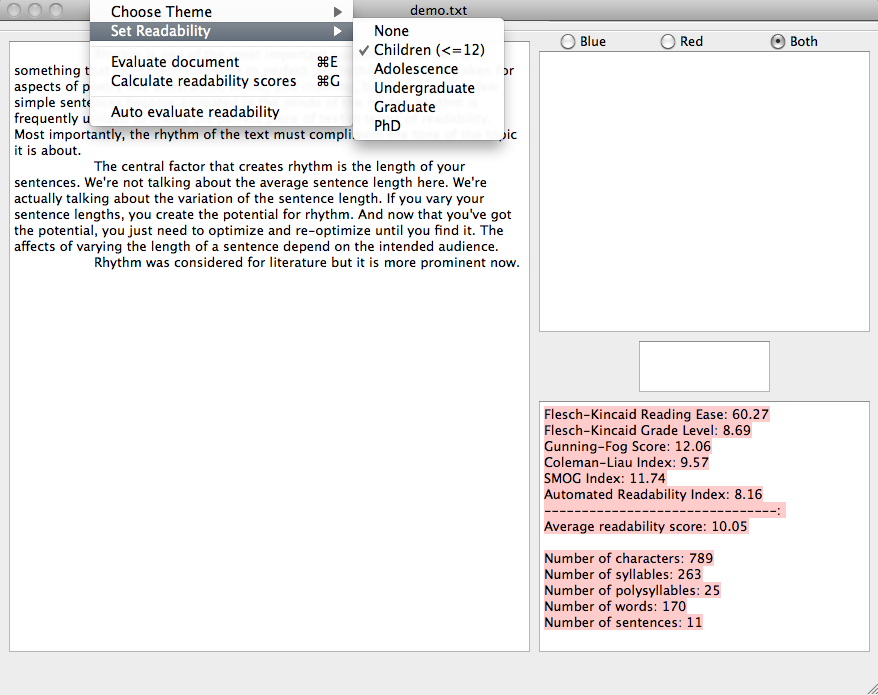
The user has completed typing a document. The application has evaluated the text and has printed comments for those sentences violating rules of style. Each comment is given a colour according to the severity of error. Also, the application gives the readability scores of the same piece of text along with statistics like number of words and sentences.



***Fig. 12.3*** *Stylistic comments and readability scores*

**12.4 Setting target readability**

If the user wishes to monitor the readability score as he types the document, he may set the target readability score. The colour of the readability box changes if the actual readability score is not in the target range.



***Fig. 12.4*** *Choosing target readability score*

**CHAPTER 13**

**CONCLUSION**

Congruent networking is a novel concept, which can be used to improve the quality of service in a particular context. Social networks have many disadvantages that can be solved by the use of congruent networks. The use of congruent networks helps us in providing the best possible people who are related to us, in that particular context. The advent of congruent networking will provide some new areas for research to be done. Some applications of this field include the following Maximizing team productivity, Assignment problem, Formation of context based Knowledge groups, Human relationship management, Comparative study of behavioral v/s Biological parameters to identify congruence, Product - Entity based congruence, Event Managed based Congruence, Pre-Market Analysis, Trend based Market analysis, Matrimonial services, Recruitment Services, E-teaching and E-learning, Graph theory based approach for relationship analytics etc.

The implementation takes into account the ego-factor of the users of the social network, and will generate an appropriate egocentric congruent network for the given context, for that user. The level of ego varies from person to person, and also varies depending on the context. Some people tend to be more egocentric during certain situations, whereas some people are more sociocentric. The congruent network will change depending on the context as well. This approach is more efficient when we consider egocentric people, that is, people who are more interested in themselves than others.

Similar contexts are implemented using the Jaccard similarity. It takes the description from the user and matches it with the description that is available in the database. In order to generate the network, the system compares the similarity of the context’s property’s description with the pools’ attributes stored in the database. The best 5 matching attributes are chosen for each of the properties for the given context. These attributes are looked for in the users’ database. The selected candidates are assumed to be congruent with the user considered. The mapping also shows an expected result, with an efficiency of 50%-80% for the users. Generation of the Egocentric congruent network takes approximately 5 to 10 seconds as it involves mapping of properties with the existing contexts, and mapping of the context with the users.

**CHAPTER 14**

**FUTURE SCOPE**

As the project comes under a new and upcoming research domain, there is a lot of scope for more enhancements as well as more research opportunities exist. Congruent networking, as a research topic, will be very helpful in the near future. Our implementation of this project has been restricted by the information collected from the various social networking sites. Due to the unavailability of time and infrastructure, we have not been able to implement this project by considering the scalability aspect. In this section some of the enhancements possible to our implementation in different sections, starting with functional improvements, followed by new approaches for the generation of congruent networks, have been listed.

**14.1 Functional Improvements**

This section lists some of the improvements that can be incorporated to the existing implementation of this project:

* Collect information from various social networking sites, including Twitter, LinkedIn, Quora, Google+ etc.
* Collect more information from Facebook which may include posts, likes, comments, etc
* Include category information provided by WolframAlpha and other services while performing similarity calculation.
* Generate a larger description for the values of the attributes in the pool database
* Use of different similarity algorithms to calculate the similarity scores and combine similarity scores of various algorithms to provide better results.
* Classification algorithms can be implemented for the matching of the pools with the context.
* Implement the context using more efficient ways.
* Implement more natural language algorithms to find the intentions behind the context.
* Generation and use of the network, using existing network generation tools, for a better visualization and better user experience.
* Use different technologies to implement the backend in a more efficient manner.
* Make the project compatible with every browser.

**14.2 New approaches**

This section lists a few new approaches that can be incorporated into our implementation.

* Host the application on the cloud and include a large number of users.
* Collect user feedback and incorporate them.
* Define the context in different ways.