Software Engineering Department

Ort Braude College

Tracking Plutchik Emotions in Series of TV Shows

In Partial Fulfillment of the Requirements for
Final Project in Software Engineering (Course 61771)

Karmiel – March 29, 2019

Authors

Rotem Shitrit 204678221

Elia Amar 205802960

Supervisor

M.Sc. Julia Sheidin

Contents

1. INTRODUCTION

In the past several years the social networks became the modern tribal bonfire of humanity. Experiences, thoughts and everyday events are constantly uploaded to the web in textual and visual forms. This stream of personal data can be recruited to analyze the views of individuals or masses, concerning almost every element of life, both for research and commercial purposes.

Microblogging, the concentrated member of social networks family, was built to keep this way of expression brief and accurate. Due to this nature, using analysis tools on microblogging messages give more accurate and insightful results, compared to other social platforms.

The TV series "Game of Thrones" has become a huge event in nowadays western culture, and its episodes are watched near live-time by dozens of millions around the world. Its unique nature and plot twists cause many discussions in social networks. Due to the variety in the series' fans communities, the discussions are very vivid and often show different points of view.

In this document, we will describe a research-oriented emotion analysis system that is suitable for short texts. It will analyze tweets that relate to the seventh season of the series "Game of Thrones", as well as the same season's line-scripts. Based on Plutchik's theory of emotions and the NRC emotions lexicon, it will classify the analysis subjects. Upon specific search parameters, it will weigh the relevant data and visualize the analysis results to the user.

2. THEORY

2.1 Domain Description

In the last couple of years TV series took over large volume of nowadays popular culture. Grand studios have made them the heir of the films industry, and the global village embraced them as the modern tribal bonfire.

Unlike former ways of entraining, the audience of TV series plays an active role as it accompanies the characters along the plots. Individual fans and communities, mostly on the web - using all kinds of social platforms, share and echo thoughts, feelings, complicated theories and jokes about many aspects of the shows they adore. In the last few years it was shown how powerful the influence of series on the real world can be, as several of them encouraged and triggered social movements or increased awareness regarding issues that were barely spoken of before.

The production studios are aware of the crowd that follows their creation, and know it is the audience's opinion they must satisfy in order to success. Not once online discussions formed a wave that was heard in a director's office and influenced specific scenes, or even caused a plot turn.

One of the most surprisingly popular genres of series is fantasy, as it was not long ago considered an isolated niche. The fantasy series became more and more common as The Lord of the Rings trilogy and Harry Potter heptalogy were played on the big screen. These two conquered the hearts of millions and were instantly added to the canon of films history.

The fantasy series Game of Thrones is watched by is the most popular show aired today. Each of its seventh season episodes was watched by approximately 23 million Americans (including via streaming and VOD platforms, excluding illegal ways). The volume of buzz around it is very hard to measure since it is discussed in many forms and over almost any kind of platform, yet over the main social networks it is approximately mentioned few million times near airing days.

2.2 Twitter as a Platform

Microblogging platforms are a short and, as implied, more one-sided version of social networks. They allow their users to express themselves within a limited range of about 100-300 characters. This limitation forces microbloggers to make succinct statements, in which they yet manage to share their thoughts. Unlike communities-oriented networks such as Facebook, microblogging sites see great value in free speech, and they mostly allow explicit or controversial statements. Therefore, they rarely censor messages, and let their users share almost any kind of idea or graphic content with their followers.

With 336 million active users in the first quarter of 2018 [1], Twitter is the most popular microblogging site in the world today, and as such it is a gold mine of data. Unlike other social platforms, almost every user's tweets are public and accessible. This option has made Twitter a steady base for social-data analysis.

Using its API, users can query for tweets within very specific time or content range. For example, the spatial specifications can limit a search to tweets from specific country or countries. Time limitation can be added so only results that were posted close to a specific date would be collected.

2.3 Sentiment Analysis Versus Emotion Analysis

Language is the main tool by which humanity expresses its feelings, thoughts and needs for thousands of years. The various ways we use it and the specific words we choose to let our emotions an out, can reveal the person that stands behind them, or at least allow a glimpse to his/her thoughts. This not-very-humble idea has inspired the focus of linguistic and psychological branches of social sciences yet was not notable enough until the age of computation.

As a field in computational social sciences, Sentiment Analysis's goal is to analyze and figure expressional tendencies in written or spoken text. It aims to generally decide whether a given text should be classified positive or negative. To reach better accuracy, in later studies another category was added, for neutral texts, as well as ones that their tendency was not clear enough.

The methods used in Sentimental Analysis include classification of words in large lexicons, in a process that involves human decisions. Usually, several people tell whether each word arouses positive or negative feeling or represents idea that can fall into these categories, and the word is added to the lexicon according to the majority's opinion. Once the lexicon is set, the analyzing system searches for positive and negative words in a given text and tells its estimated nature according to the positive-negative ratio.

The understanding that the human mentality is not binary, but based over a set of basic emotions, made it necessary to analyze text in a different way, that could detect specific and more subtle tendencies. Several social researches concluded that about eight main emotions compose our mental perception. A prominent theory in this field was developed by R. Plutchik [2], and it discusses the nature of emotions and their role in the evolvement of creatures. Emotion Analysis combined its preceding system's methods with these core emotions, to create extended lexicons that will fit several categories of emotions. The human process of sorting the words is done in the same way, though close emotions are commonly allowed to overlap, as they share same words. An analyzing system that works over this kind of lexicon, produces a rate for each emotion and classifies the examined text according to the most dominant emotion.

2.4 Different Approaches to The Study of Emotions

At some level, emotions used to be taken for granted by humanity, for their role in our lives is very fundamental. Before C. Darwin published his theory in 1872 [3], the scientific discussion regarding this field was very different from the way we know it today. His understanding of emotions was that they are discrete (modular), as he distinguished them from one another based on their origins in facial expressions. While many scientists, such as P. Ekman and V. Friesen [4], followed his way, W. Wundt [5] proposed a different approach in 1896. To his perception, emotions differentiate via two dimensions: low-high and pleasant-unpleasant. Hence, following his logic, what we might see as separate emotions, are more basic ones with different values of pleasantness and intensity. Later researchers, such as R. Plutchik and H. Schlosberg [6], adhered to the dimensional option, but used it under different interpretations.

W. Wundt also suggested a third way, that considers both ideas and sees emotions as modules that vary along scales. Along the years, more scientists got closer to this theory, though the differences between them yet remain very noticeable. Their points of views affect their way of study, as well as the applications that are based on their works. For example, most of the projects that use facial expressions to recognize emotions, stick to the first theory, while others who wish to focus on more global aspects of emotions, tend to use Plutchik's works.

2.5 Plutchik's Theory of Emotions

One of the most profound theories that tried to explain the structure of emotions, alongside their natural sources, was proposed in 1980 by R. Plutchik. The theories of emotions that were published prior to his work, did not discuss important aspects of emotions, such as their evolutional functions and the fact that most evolved creatures share them. His research was based on the understanding that emotions are essential product of primal nature, as they should, upon life-risking situations, trigger certain behavioral or biological responses that will increase the chances of survival. With this resolution he analyzed emotions with the common psychoanalysis and social paradigms.

Plutchik suggested eight emotions to be considered primary emotions: joy, trust, fear, surprise, sadness, disgust, anger and anticipation, as the last four respectively contrast the first four. According to the theory, the rest of the emotions we feel derive from these basic emotions, as they are secondary forms of the original ones. For example, rage would be considered an escalated anger and annoyance is just a shallow type of anger. Given this logic, he presented both mild and intense levels for every primary emotion, to create a set of sixteen emotions.

The structural model that Plutchik developed places the basic emotions in a circle, that represents the relations between them. The closer two emotions are to each other in the circle, the similar they are. As a result, contrast emotions were set against one another. The theory explains that the emotions that are not concluded within the set of sixteen, are the product of combinations within the set. Integrations of non-contrasting pairs that share the same level of intenseness create a new emotion. When these mixtures of emotions, dyads, are done between adjacent emotions in the model, they are called primary dyads. More emotions can be represented as combinations of distanced emotions in the same level of intensity. This wheel of basic emotions and their dyads can be seen in fig. 1.

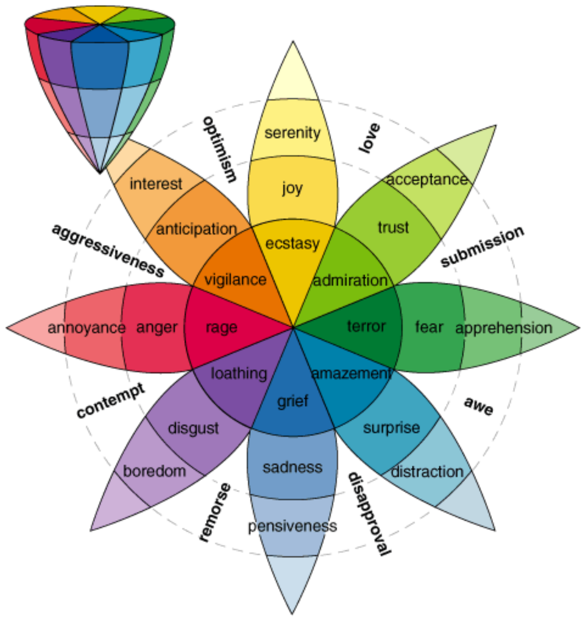


Figure 1: The basic emotions in Plutchik's theory (inside the wheel) and their primary dyads (around the wheel), as they respectively form by a combination of adjacent pairs

2.6 Effective Visualization of Emotions

All the significant theories that discuss emotions are based upon a set of emotions, composed of at least five prime emotions. When trying to express insights in a complex structure that contains several equal pillars, it is important to choose a way of visualization, so the data will be reflected clearly. The classic approach makes use of simple models. Besides basic diagrams such as bar and pie charts, other models that are used in later research seem to be more informative in some cases, as they focus on the relations within the set, and less on other factors.

Due to their simplicity, the traditional bar and pie charts are commonly used in emotions analysis papers, such as in S. Mohammad's paper about tracking emotions in novels and fairy tales [7]. The bar chart from the paper can be seen in fig. 2. It lets the reader see the relations between the measured emotions. The pie chart from the paper, as appears in fig. 3, also shows the volume that each emotion holds, with comparison to the whole. These charts are usually used to give basic insights and general impression about the data.

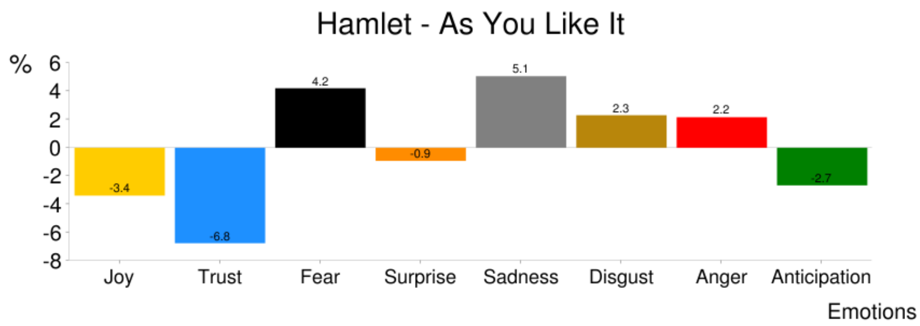


Figure 2: A bar chart used to compare the analyzed emotions from the texts of Shakespeare's plays "Hamlet" and "As You Like It"

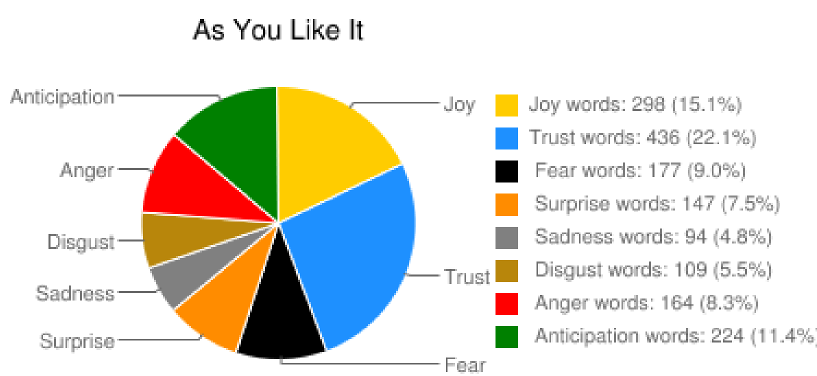


Figure 3: A pie chart used to view the emotions analyzed from the text of Shakespeare's play "As You Like It"

The radar chart model is a developed form of the bar chart. The original side-by-side standing bars were replaced with spikes that stand in a circle, with one common starting base. The scale looks like a net that fits each individual spike. This formation helps the viewer to compare the volume of each element with the other elements, and to easily see which elements are more dominant than the others. This type of visualization is widely used, both in emotion analysis articles and applications. fig. 4 shows how it was used in the "EmpaTweet" project regarding several topics [8].

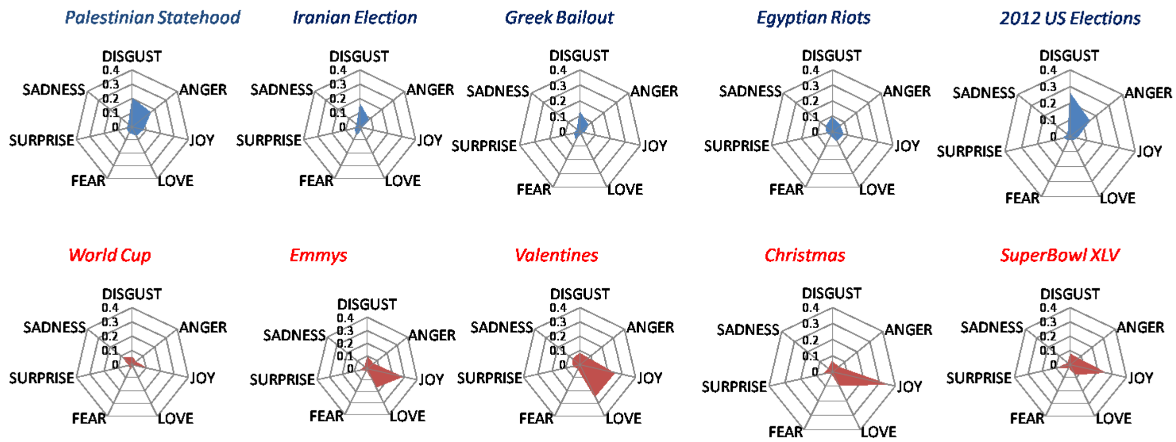


Figure 4: Radar charts used to visualize emotions regarding several controversial issues, as analyzed

Another idea that is used to visualize emotions is coloring the previous mentioned diagrams. This method gives each emotion its own color. If a large set of emotions is shown, the colors can be matched, in a way that close emotions are given close colors. This kind of visualization was used in “EmotionWatch” [9]. In the project’s paper, the spikes of a circular radar chart were colored with the color of the most dominant emotion. As can be seen in fig. 5, this project was based on a large set of twenty emotions. Hence, emotions of the same group were adjacent and given different shades of the same color.

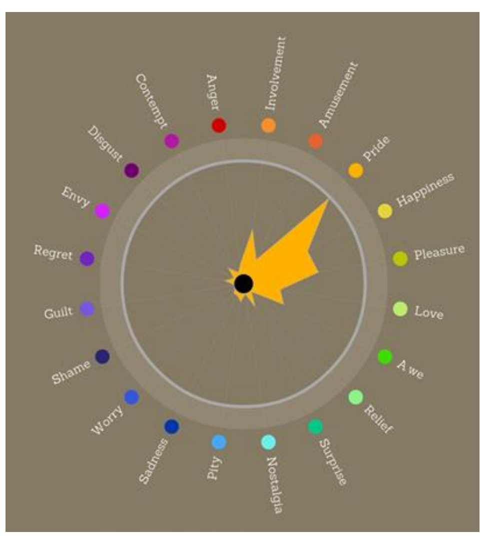


Figure 5: A circular radar chart that shows groups of close emotions with similar colors. All the spikes of the chart are colored with the color of the leading emotion.

2.7 Text Analysis

With the rapid evolution and everyday usage of social media, a sophisticated analysis of text has become a major and important research area. It has become necessary to develop tools that would be able to comprehend successfully this flood of data, in order to get make it useful for research, social, economic fields. Different types of texts, such as short texts, scientific texts and everyday-language texts should be analyzed differently, due to differences in their nature. Hence, it would be wrong to analyze one type, with tools that are meant for the others. Also, it would not be smart to ignore the unique features of an examined text, since they can be used to get a more accurate analysis result.

In our project we gather the information regarding the emotions of the audience about TV series, beside analyzing the emotions that the series' scripts express. In order to analyze the audience's opinion and the script-lines, regarding the features of the examined texts, we use the following tools for their analysis.

2.7.1 N-Grams

An n-gram is a contiguous sequence of n items from a given sample of text or speech [10] . In our project we use the 2-gram model (bigram or digram) to analyze the words in pairs. In this way we can detect the context of the term, for example, negation context. Bigrams are used in most successful language models for speech recognition. They are a special case of N-gram.

Bigram frequency is one approach to statistical language identification. The distribution of a frequency of a bigram in a given string can be used for computational linguistics as well as other textual and statistical analysis. Bigrams can provide conditional probability for a token, given its preceding token. This conditional probability relation is described by the following formula:

$$P(w_n \mid w_{\mathrm{n}-1}) = \frac{P(w_{\mathrm{n}-1}\,, w_n)}{P(w_{n-1})}$$

Where W_n is the current word, W_{n-1} is the previous word, P(x) is the probability function of x, P(x|y) is the conditional probability of x given y, and P(x,y) is the probability of the cooccurrence of x and y.

2.7.2 Linear Regression

Linear Regression is a simple machine learning model for regression problems, i.e., when the target variable is a real value. It is a math approach to find the parameters of the assumed linearly relationship between a dependent variable and one or more independent variables. The case of one independent variable is called simple linear regression. For more than one independent variable, the process is called multiple linear regression. In our case, we use the multiple linear regression model. The main purpose of multiple linear regression is to model the linear relationship between the explanatory (independent) variables and response (dependent) variable:

$$Y = a_1X_1 + a_2X_2 + \cdots + a_nX_n + b$$

by finding the parameters: a_i the independent contribution of each independent variable X_i to the value of the dependent variable Y, and b is constant (which include the error term).

In the case of multiple linear regression, instead of our prediction being a line in 2-dimensional space (simple linear regression), it is a hyperplane in n-dimensional space.

For example, in 3D, our plot would be a plane, as can be seen in fig. 6:

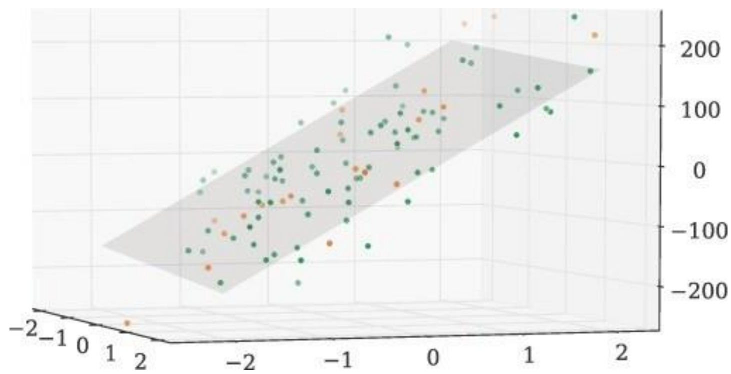


Figure 6: A of multiple linear regression in 3D gives a plane

Multiple linear regression is a function that allows to analyst to make predictions about one variable base on the information that is known about another variables. It can only be used when there is a relationship between the dependent variables and the independent variables.

The independent variables are not too highly correlated with each other. Y_i observations are selected independently and randomly from the population.

Different values of the weights $(a_1, a_2, \ldots a_n)$ gives us different lines, and our task is to find weights for which we get best fit. For this, we measure how close the Y, that we get from a set of weights, to the target value. We measure it by calculate the mean square error:

$$J(a) = \frac{1}{n} * \sum_{i=1}^{n} \left(Y(X_i) - Y_{target}\right)^2$$

2.8 The Algorithm

The system runs the analysis over two types of data; tweets and scripts. Both to be provided by the system's manager. The system runs different kind of emotion analysis on each of the data types. The analysis is done with respect to the NRC (National Research Council Canada) emotions lexicon (2013) [11], which follows Plutchik theory of emotions. Due to the nature of the data, the system conducts the analysis in a way that is suitable for short texts.

The main parts of the process are described in the chart below (fig. 7).

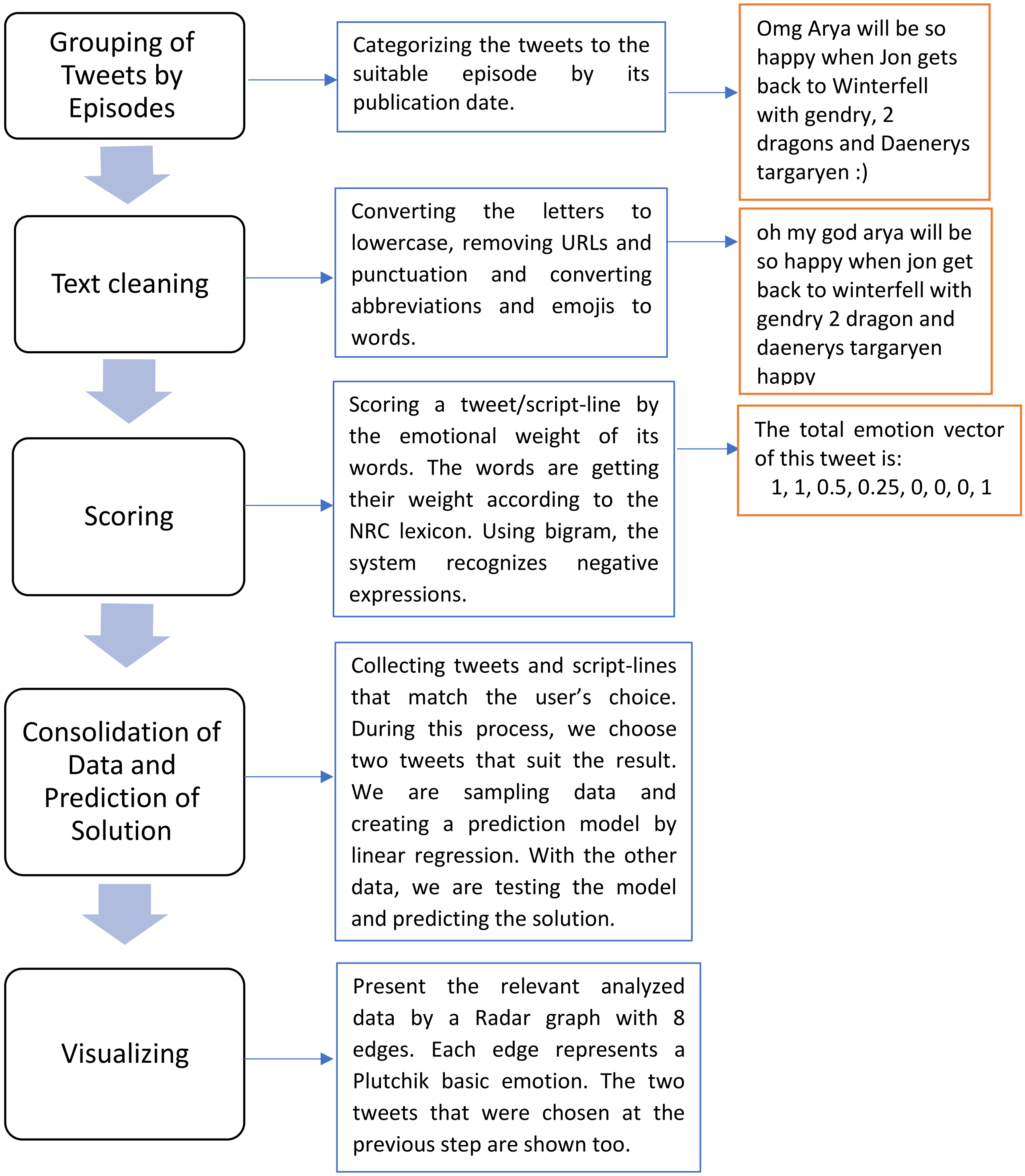


Figure 7: The main processes of the system. Described from the top down, each process (left column) is described by a short explanation (middle) and a concurring example (right), if relevant.

2.8.1 Grouping of Tweets by Episodes

After we got the data in the database, the tweets were categorized to the suitable episode by their publication dates. If a tweet's publication date was between two episodes airing dates, it got the earlier episode.

2.8.2 Text Cleaning

Based on the instances in all the tweets, a table of common significant strings was set in a human decision process, prior to the analysis. The system used it to replace the common strings that the lexicon could not handle, such as hashtags, emojis and abbreviations. Each tweet in the group was cleaned of not-relevant strings, such as URLs, punctuation, prepositions etc. A tweet was deleted in case that its language was not English. The words were stemmed and lemmatized, in order to reduce them to their original form.

We decided to handle retweets as original ones, as we came to understand that if users chose to retweet, they identified with the emotions that were expressed in the tweet. We didn't find it necessary to handle cases of duplications of tweets, due to the negligible amounts of them in files we examined.

As for the scripts, every script-line was handled separately. We broke them into words after cleaning them of tokens that were not letters. The words were stemmed and lemmatized, in order to reduce them to their original form.

2.8.3 Scoring

From this part, the process of the tweets and script-lines was the same. We set an emotions vector, that represents the basic emotions of Plutchik's theory. Each word in the tweet/script-line was compared to the lexicon. According to the sentimental tendency of the word, the emotions vector was increased at the relevant cells. Using the bigram model, the system recognized negative expressions, and updated the emotion vector accordingly. In cases of complex sentimental tendency, the word was compared to analysis results of tweets/lines that it appeared on. To get the accurate context, the comparison was done within the same group of episode/script. When we got the emotion vector of each word, we scored all the vectors, according to the following formula:

$$V_{score} = \sum_{i=0}^{n} V_{word}$$

Finally, we normalized the emotions vectors, so that each tweet/script-line had equal volume in the result. After the normalization, the emotion vector included values between 0 to 1. The normalization was done with the following formula. *Zi* and *Xi* respectively stand for the normalized and original values of the vector, while *max(X)* stands for the maximal value in the original vector.

$$Zi = \frac{Xi}{max(X)}$$

2.8.4 Consolidation of Data and Prediction of Solution

The saved results of the analysis summed only upon search. Users chose their search parameters and selected one episode they wish to examine. The system found the relevant tweets and summed their normalized emotions vectors. The same process was done with the lines of the relevant script.

During the consolidation of the result from the relevant tweets, the system used linear regression to build a model that aimed to predict the emotions vectors of new examined tweets. We built eight models, one for each emotion. The criteria we provided to the models were the words of the.

Each model was built fifteen times; Each with different minimal number of occurrences, between one to fifteen, that a word from the lexicon was required to appear in the training tweets, in order to have a weight in the model. The results were taken from the model that was found to be the most accurate one, based on the minimum mean square error of the models. Using the targets of the training data, the model modified the weights of the criteria. To keep the system runtime at a fair range, the training data held 80% of the gathered tweets, in a way that if more than 1,000 tweets were gathered, only 800 of them would be used as training data. If less than 400 tweets were gathered, the system didn't initiate a prediction at all, due to too few data.

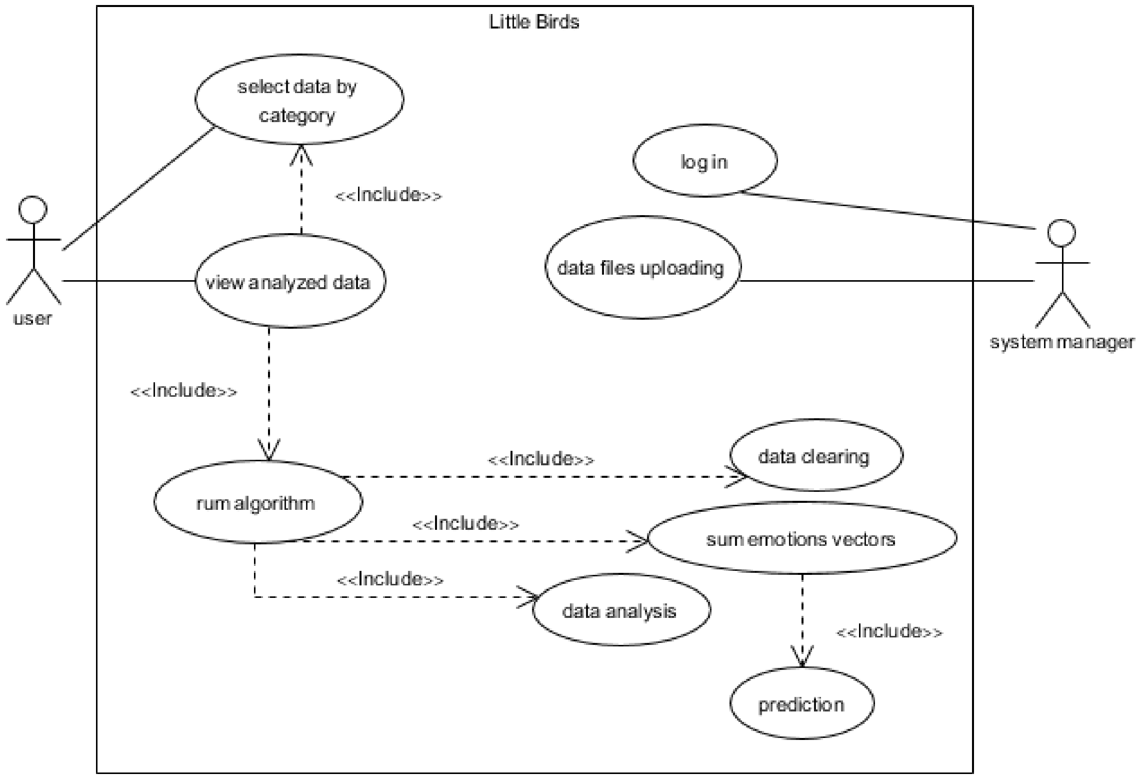
Then, the models received the testing data. It held the rest 20% of the gathered tweets, in a way that if more than 1,000 tweets were gathered, only 200 of them would be used as testing data (As mentioned, if less than 400 tweets were gathered, the system didn't initiate a prediction at all due to too few data). Using the testing data, the models predicted the final value of each emotion. Combined, the models gave a whole predicted emotions vector.

2.8.5 Visualizing

Eventually, the system visualized the results in a comparable view of radar chart, since it is widely used for conveying emotion analysis information. We use the radar chart to show the eight Plutchik basic emotions, each represented by one of the radar edges. In addition, the system presented two tweets that were found close to reflect the detected trend of the result. That was meant to provide a glimpse to the analysis process as well as to allow the user to "feel" the processed data.

3. SOFTWARE ENGINEERING DOCUMENTS

`3.1 Requirements



UC1: Select data by category

The user selects an episode and a category from 3 options in order to retrieve data for analysis.

Goal: Get the user category.

Preconditions: The database is not empty.

Pseudo code Flow:

Actor	System
1) Click on "Start" button	1a) Display episodes combo box.
2) Select an episode	2a) Display 3 categories
3) Choose a category	4) Pull the relevant data from DB

UC2: Log in

The system manager login to the system

Goal: System manager login to the system.

Preconditions: The system manager details exist in the DB

Possible errors: Enter incorrect details.

Limitations: Correct details.

Pseudo code Flow:

Actor	System
1) Enter username and password	2) Verify the details
3) Get into the system	

UC3: Data clearing

The system clears the tweets and converts the words to their basic form in order to be capable to analyze them.

Goal: Clean the tweets and convert the words to their basic form.

Preconditions: The database is not empty.

Limitations: Cannot translate foreign language tweets

Pseudo code Flow:

Actor	System
	1) Collect all the tweets from the DB
	2) for each tweet:
	2.1) remove punctuations
	2.2) convert emojis to emotions
	2.3) split to words
	2.4) convert each word to its basic form
	2.5) merge all words to a sentence
	2.6) save the sentence in cleantext variable.

UC4: Data analysis

The system analyzes all the clean text tweets and get an emotion vector which reflect the emotion of a specific tweet.

Goal: Get an emotion vector for each tweet.

Preconditions: Tweet passed the cleaning process

Possible errors: Incorrect total emotion vector

Limitations: Cannot recognize cynicism

Pseudo code Flow:

Actor	System
	1) for each tweet:
	1.1) initialize an emotion vector to zeros
	1.2) break clean text to words
	1.3) each word gets an emotion vector
	1.4) sum all the emotion vectors
	1.5) save the sum in a variable emotion vector

UC5: Data files uploading

The system manager uploads a csv file in order to update the DB.

Goal: Upload data file.

Preconditions: System manager logged in to the system

Possible errors: Duplicates data.

Limitations: Can upload only one file and only csv file.

Pseudo code Flow:

Actor	System
1) Drag a csv file to the window	
2) Drop the file into the window	
3) Click "upload" button	4) Check the conditions of the file
	5) Update the database

UC6: Run algorithm

Run the algorithm by the category that was chosen by the user.

Goal: Run the algorithm successfully

Precondition: User choose a category

Pseudo code Flow:

Actor	System
	1) Data clearing
	2) Data analysis
	3) Sum emotions vectors

UC7: Sum emotions vectors

Sum all the relevant emotion vectors and get a total emotion vector.

Goal: Get total emotion vector which reflect the emotion of the audience.

Preconditions: Tweets went through the analysis process

Pseudo code Flow:

Actor	System
	1) Collect the relevant tweets from DB
	2) Initialize a total vector to zeros
	3) For each relevant tweet:
	3.1) Add the vector emotions to the total vector
	3.2) If we have passed on 90 percent from the tweets:
	3.2.1) Call the prediction function
	3.2.2) Save 2 tweets that are similar the total vector

UC8: Prediction

Predict the emotion vector of new tweet by linear regression models.

Goal: Predict as much as close the emotions vector of a tweet.

Preconditions: Tweets went through the analysis process

Pseudo code Flow:

Actor	System
	1) Get relevant tweets from DB
	2) For each emotion:
	2.1) Create csv file
	2.2) Set criteria by words from the lexicon that include the emotion
	2.3) Set rows of csv by the tweets
	2.4) For each tweet:
	2.4.1) Increase for each word in tweet the suitable word of the criterion (if exist)
	2.5) For t=1 to 15:
	2.5.1) For each criterion:
	2.5.1.1) Delete criterion if the sum of the number of occurrences smaller than t
	2.5.2) Create linear regression model with 80% of the date (training data)
	2.5.3) Predict the other 20% data (testing data)
	2.5.4) Save the best minimum mean square error and the t which give it

UC9: View analyzed data

The system presents the analyze data by the user choice

Goal: Presents a radar graph which reflect the expressed emotions in tweets or in scripts

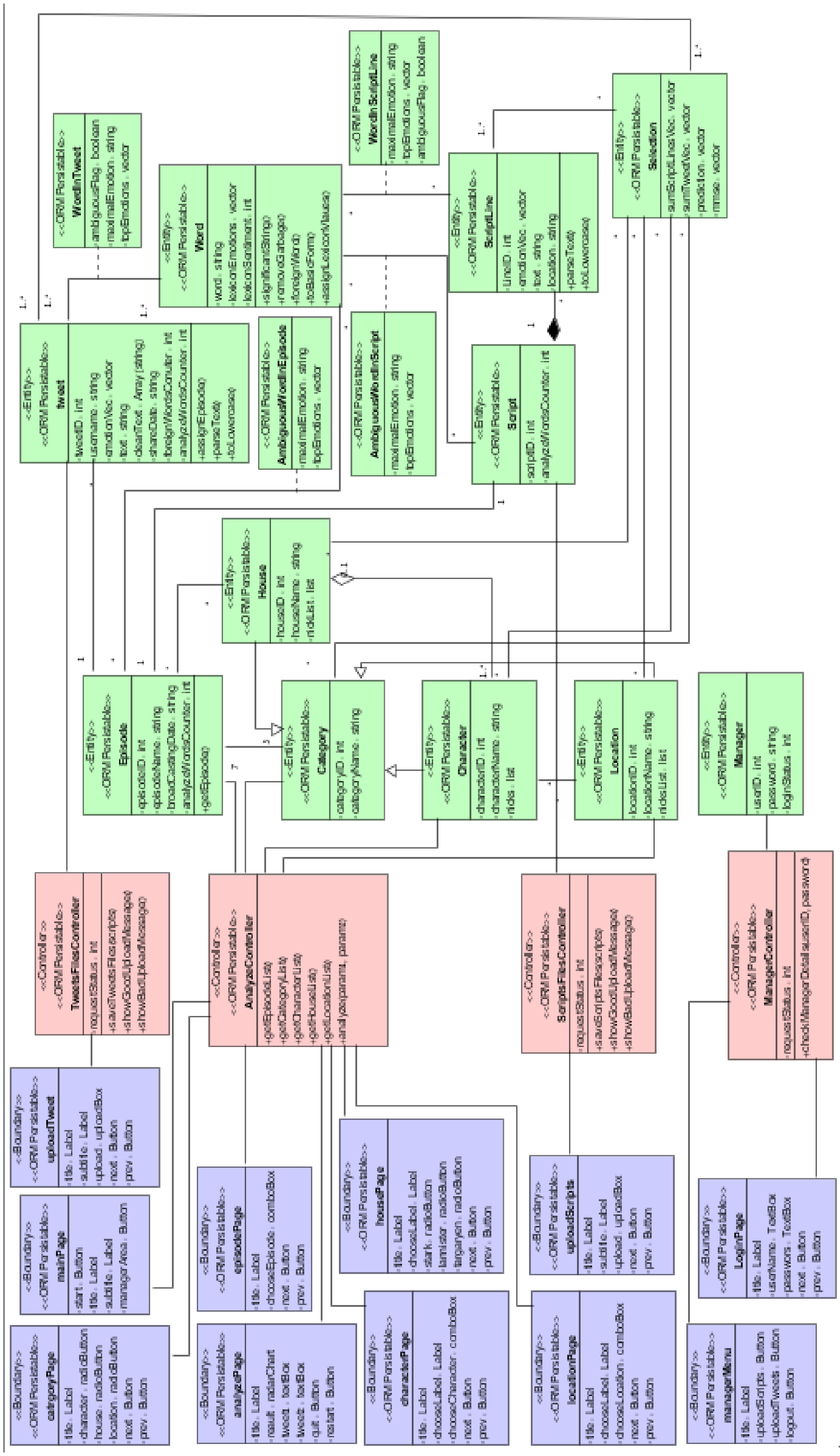
Preconditions: Tweets went through the summing process and the system receive two similar tweets

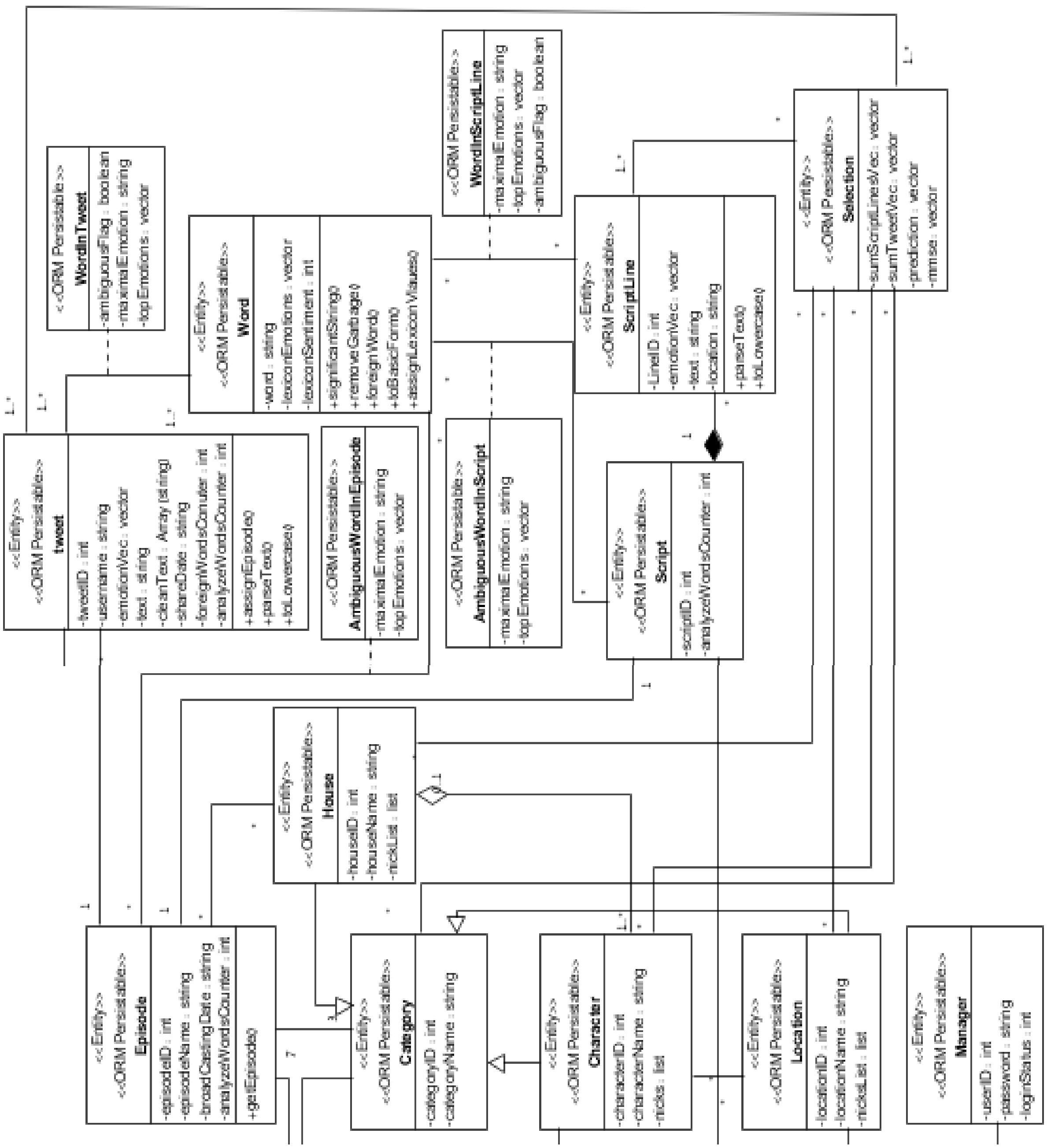
Pseudo code Flow:

Actor	System
	1) Present a radar graph and similar tweets

3.2 Design

3.2.1 Class

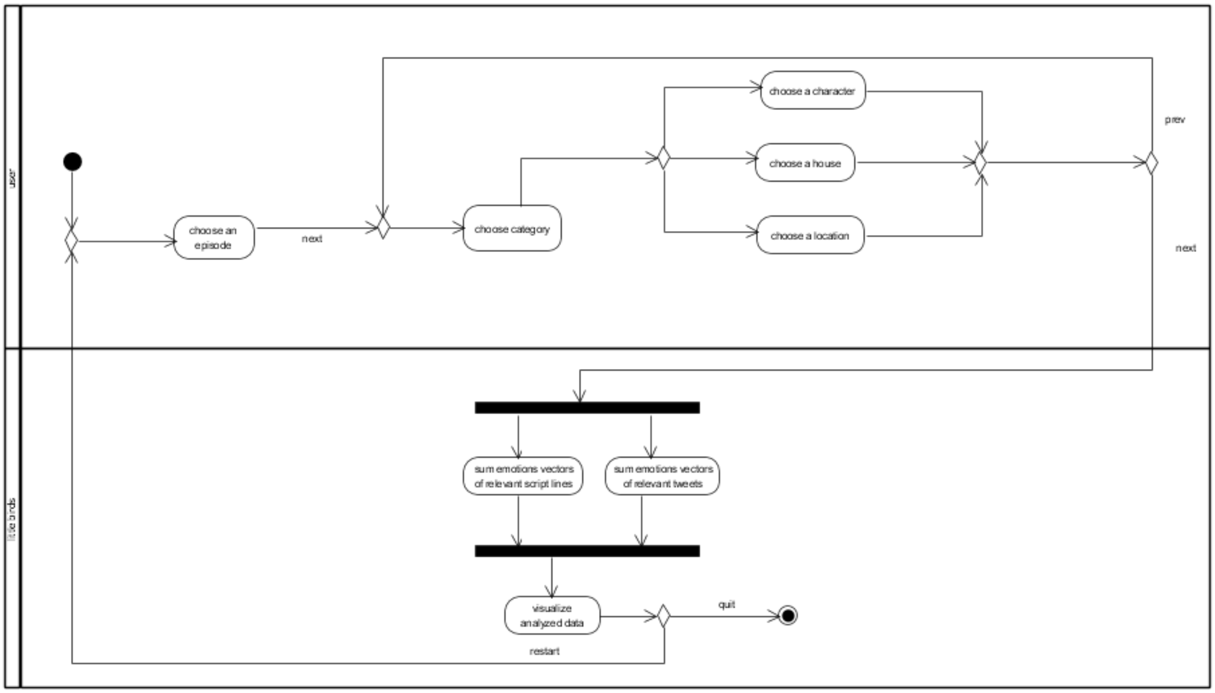




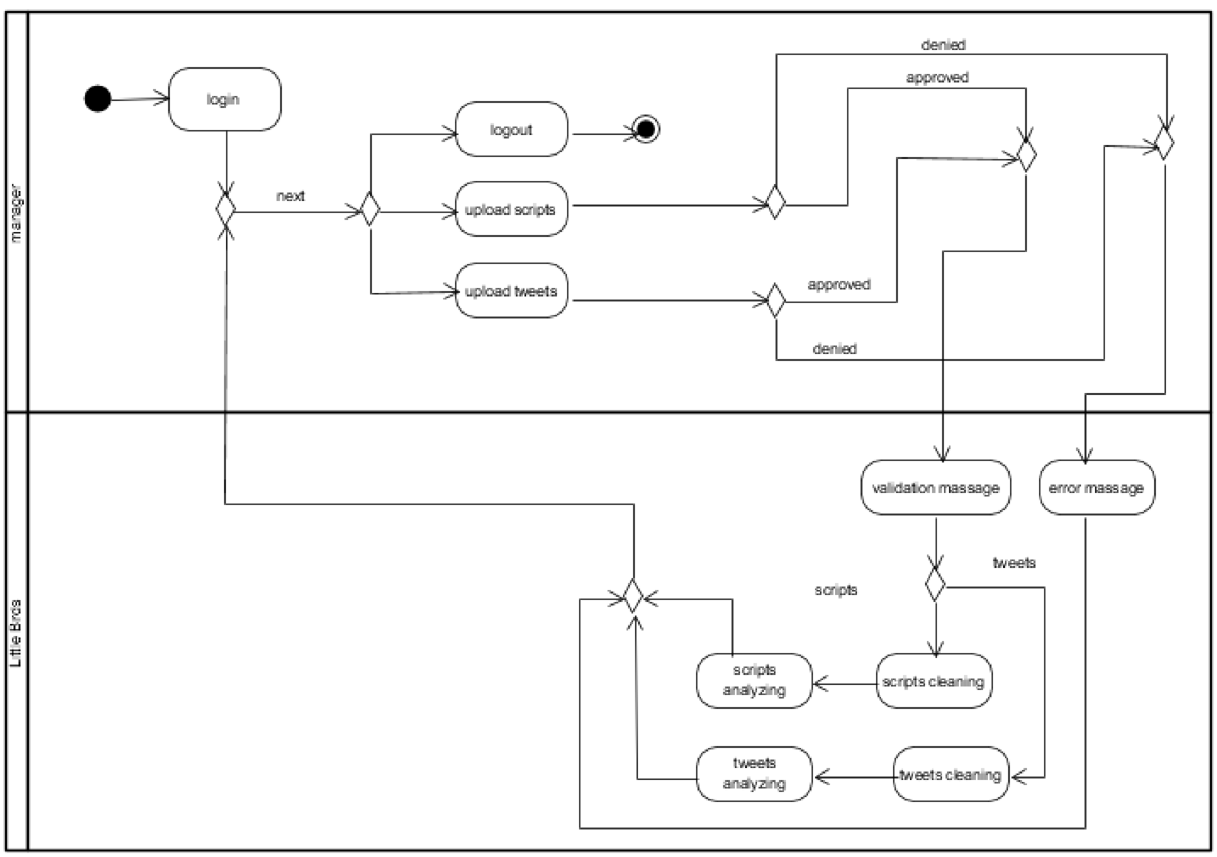
<<Entity>>
<<ORM Persistable>>
tweet
-tweetID : int
-username : string
-emotionVec : vector
-text : string
-cleanText : Array (string)
-shareDate : string
-foreignWordsCounter : int
-analyzeWordsCounter : int
+assignEpisode()
+parseText()
+toLowercase()
<<ORM Persistable>>
WordInTweet
-ambiguousFlag : boolean
-maximalEmotion : string
-topEmotions : vector
<<Entity>>
<<ORM Persistable>>
Word
-word : string
-lexiconEmotions : vector
-lexiconSentiment : int
+significantString()
+removeGarbage()
+foreignWord()
+toBasicForm()
+assignLexiconValues()
<<ORM Persistable>>
AmbiguousWordInEpisode
-maximalEmotion : string
-topEmotions : vector
<<ORM Persistable>>
AmbiguousWordInScript
-maximalEmotion : string
-topEmotions : vector
<<ORM Persistable>>
WordInScriptLine
-maximalEmotion : string
-topEmotions : vector
-ambiguousFlag : boolean
<<Entity>>
<<ORM Persistable>>
ScriptLine
-LineID : int
-emotionVec : vector
-text : string
-location : string
+parseText()
+toLowercase()
<<Entity>>
<<ORM Persistable>>
Script
-scriptID : int
-analyzeWordsCounter : int
<<Entity>>
<<ORM Persistable>>
Selection
-sumScriptLinesVec : vector
-sumTweetVec : vector
-prediction : vector
-rmse : vector
<<Entity>>
<<ORM Persistable>>
House
-houseID : int
-houseName : string
-nickList : list
<<Entity>>
<<ORM Persistable>>
Episode
-episodeID : int
-episodeName : string
-broadCastingDate : string
-analyzeWordsCounter : int
+getEpisode()
<<Entity>>
<<ORM Persistable>>
Category
-categoryID : int
-categoryName : string
<<Entity>>
<<ORM Persistable>>
Character
-characterID : int
-characterName : string
-nicks : list
<<Entity>>
<<ORM Persistable>>
Location
-locationID : int
-locationName : string
-nicksList : list
<<Entity>>
<<ORM Persistable>>
Manager
-userID : int
-password : string
-loginStatus : int

3.2.2 Activity

User activity:



Manager activity:



3.2.3 GUI

The system will allow two main action-channels. Both channels will start in the system's home **screen 1** (see fig.8). In this explanation we named the simple right facing arrow 'Next', and the left U turn arrow 'Back'. At each screen, the user can open a help page by clicking on the question mark at the upper right corner.

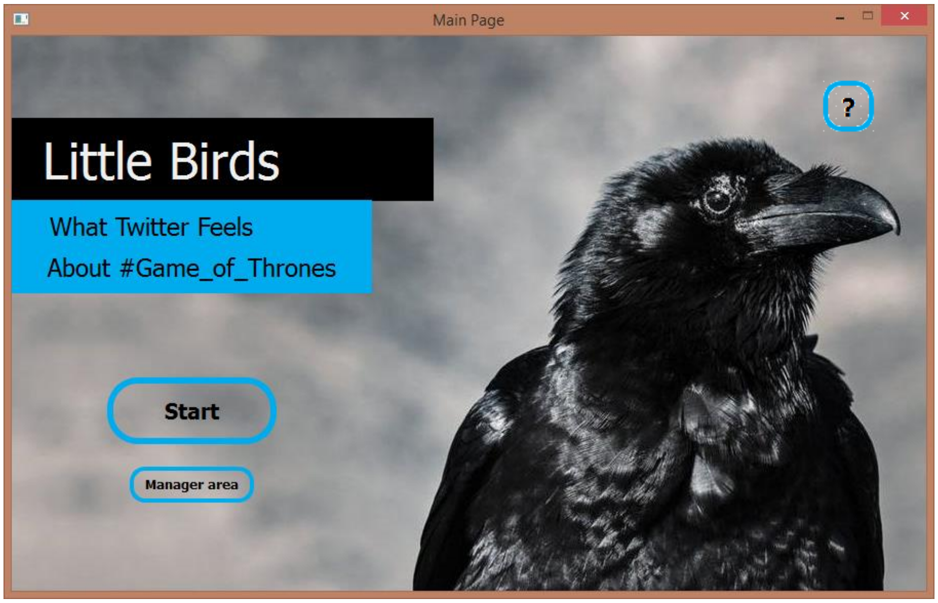


Figure 8: ***Screen 1*** *- the system's home screen. Clicking on "start" will get the user to the search channel. Clicking on "Manager area" will get the user to the data upload channel*

Navigating to the first channel – the search option - will be done by clicking on 'Start' in the home screen (see fig.8). This option will be open to all the users. Following this path, the system will show three consecutive selection screens. In the first screen, screen **1.1.1** (see fig. 9), the user will be asked to select which episode the search will run with regards to. The second screen, **screen 1.1.2** (see fig. 10), will be dedicated to choosing a search-category.

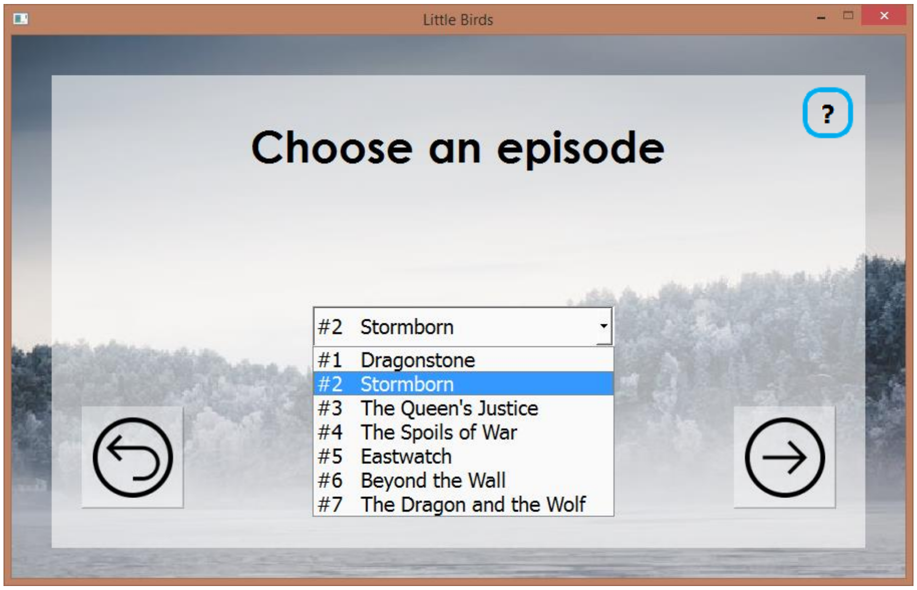


Figure 9: ***Screen 1.1.1*** *– the first search-parameter selection screen. Clicking on "Back" (left button) will get the user to the home screen (**1**). Clicking on "Next" (right button) will get the user to* ***screen 1.1.2***

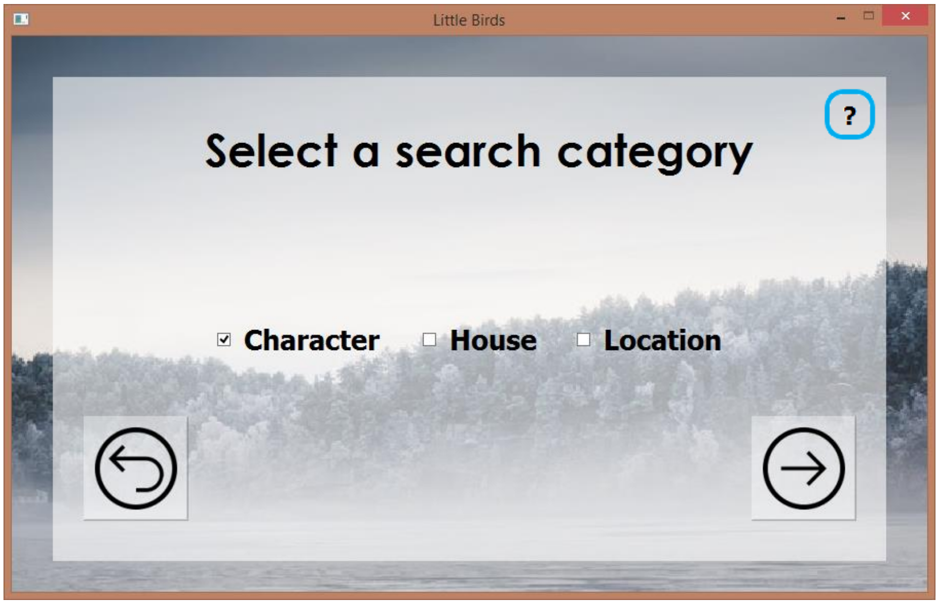


*Figure 10: **Screen 1.1.2** – the second search-parameter selection screen. Clicking on "Back" will get the user to **screen 1.1.1**. Clicking on "Next" can get the user to the last search-parameter selection screen, which will be decided based on the selected category in this screen*

The third selection screen will vary, as it will depend on the category that the user chose in **screen 1.1.2**. If the user chose to search by a **character, screen 1.1.3.1** (see fig. 11) will be shown.; Alternatively, in a search by a **location, screen 1.1.3.2** (see fig. 12) will be shown. Following this pattern, as search by a **house, screen 1.1.3.2** (see fig. 13) will be shown. In each of these three screens, the user will specify the exact character/location/house (respectively) by which the search will be done.

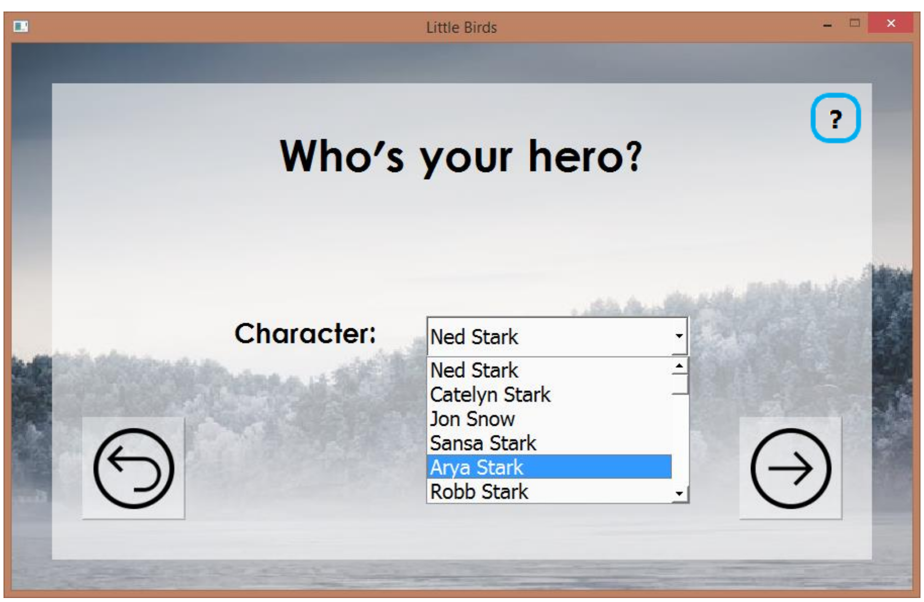


*Figure 11: **Screen 1.1.3.1** – the first option for the last search-parameter selection screen. Followed by **screen 1.1.2** in case the chosen category was **character**. Clicking on "Back" will get the user to **screen 1.1.2**. Clicking on "Done" will get the user to **screen 1.1.4***

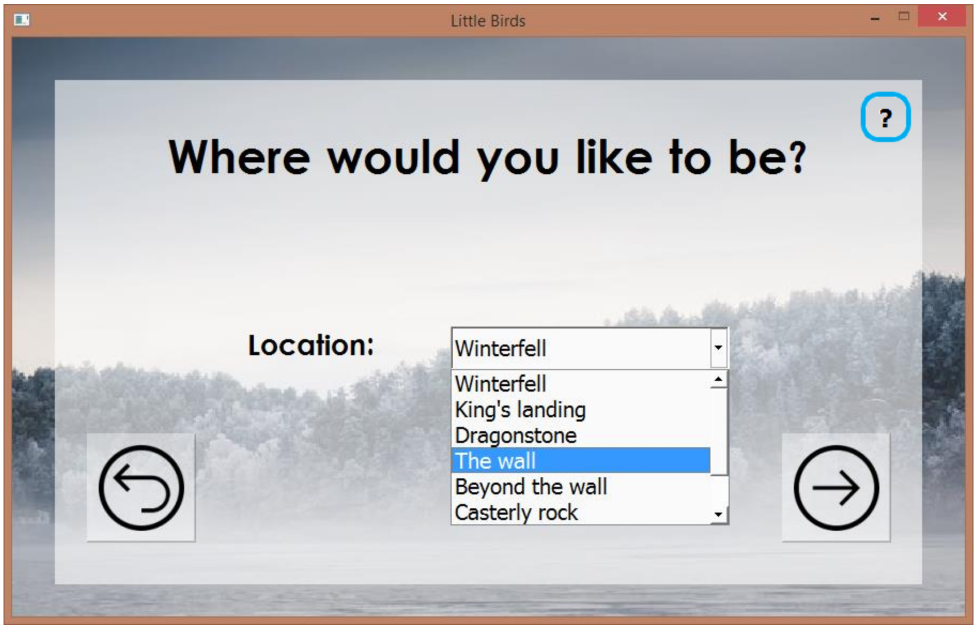


*Figure 12: **Screen 1.1.3.2** – the second option for the last search-parameter selection screen. Followed by **screen 1.1.2** in case the chosen category was **location**. Clicking on "Back" will get the user to **screen 1.1.2**. Clicking on "Done" will get the user to **screen 1.1.4***

*Figure 13: **Screen 1.1.3.3** – the third option for the last search-parameter selection screen. Followed by **screen 1.1.2** in case the chosen category was **house**. Clicking on "Back" will get the user to **screen 1.1.2**. Clicking on "Done" will get the user to **screen 1.1.4***

After getting all the search parameters, the system will view the analysis results in **screen 1.1.4** (see fig. 14). The relevant analysis results will be visualized using a radar chart. The chart will be accompanied with two tweets, that reflect the significant emotions in the analysis' result. This is the final screen in the search channel.

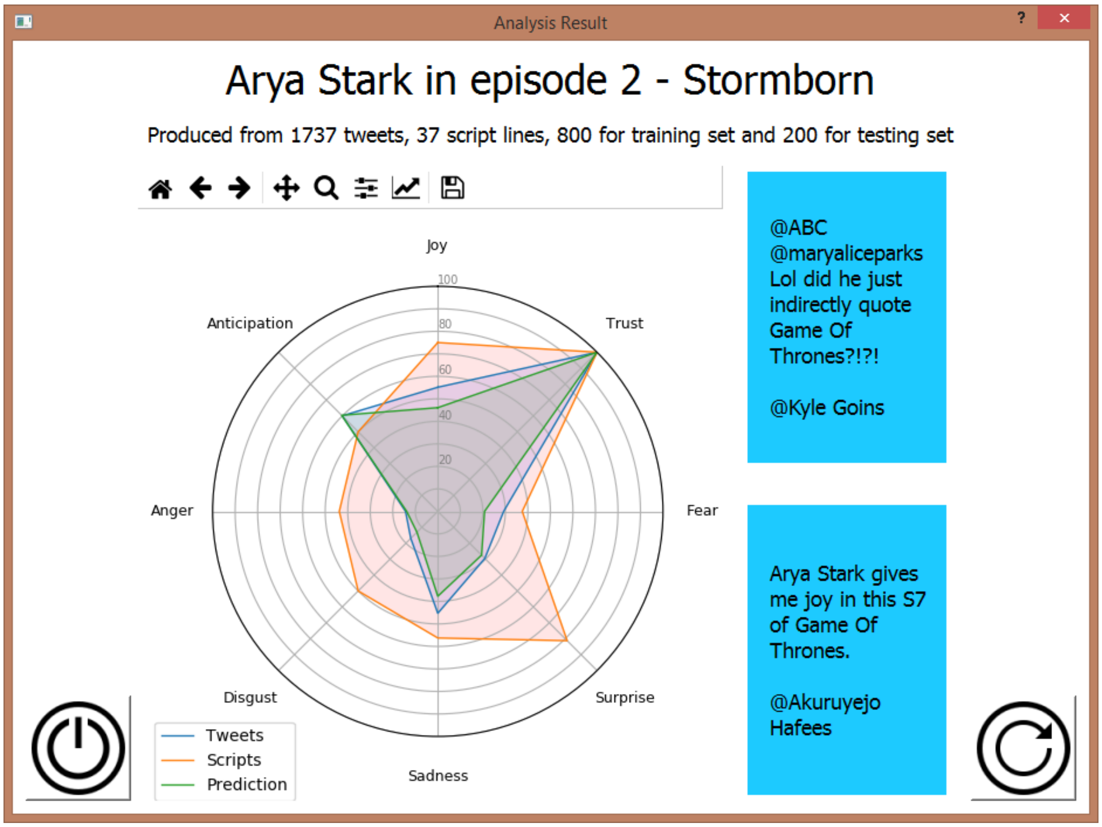


*Figure 14: **Screen 1.1.4** – the analysis results screen. Followed by either of the screens **1.1.3.1**, **1.1.3.2** and **1.1.3.3**. Clicking on "Quit" (left button) will terminate the system. Clicking on "Restart" (right button) will get the user to the home screen **(1)**.*

Navigating to the second channel – the data upload option - will be done by clicking on 'Manager area" in the home screen (see fig.8). In order to proceed in this option, the user must be identified as the system's manager. Hence, a login action will be asked in **screen 1.2.1** (see fig. 15).

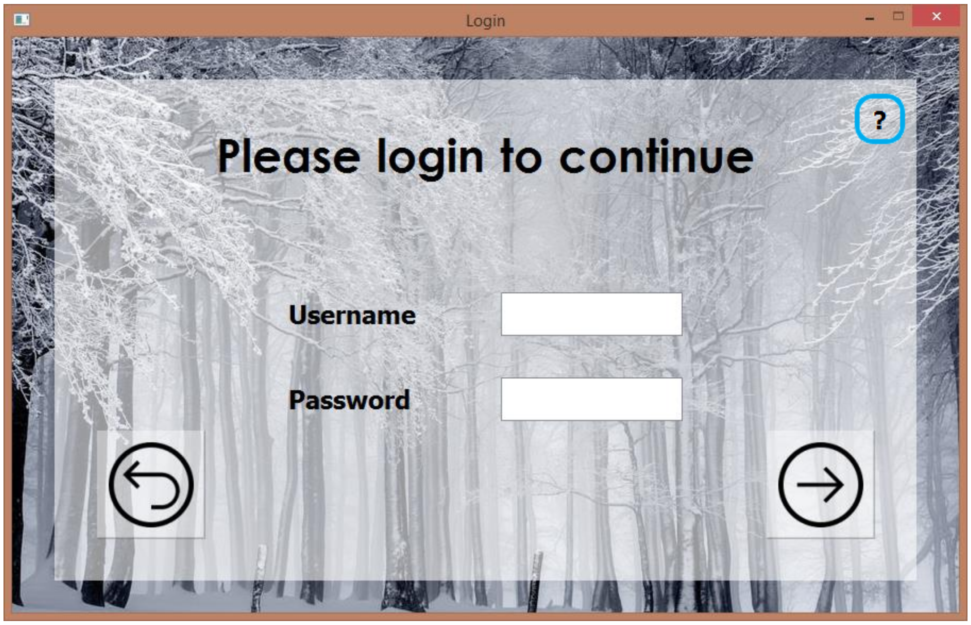


*Figure 15: **Screen 1.2.1** – the manager's login screen. Clicking on "Back" will get the user to the home screen (**1**). Clicking on "Next" will get the user to **screen 1.2.2** only if shown the user's details will be verified. In case they will not, an error message will be*

Upon successful login, the system will show the manager's menu screen – **screen 1.2.2** (see fig. 16). In this menu the manager can choose either one of two options. The first option is to upload to tweets files (as .csv files). Doing so will lead to **screen 1.2.3.1** (see fig. 17). Later, this option will be continued by a request to upload scripts files (as .txt files)' which will happen in **screen 1.2.3.2** (see fig. 18). The second option, logout, will log the manager out of the system and show the home screen (**1**).

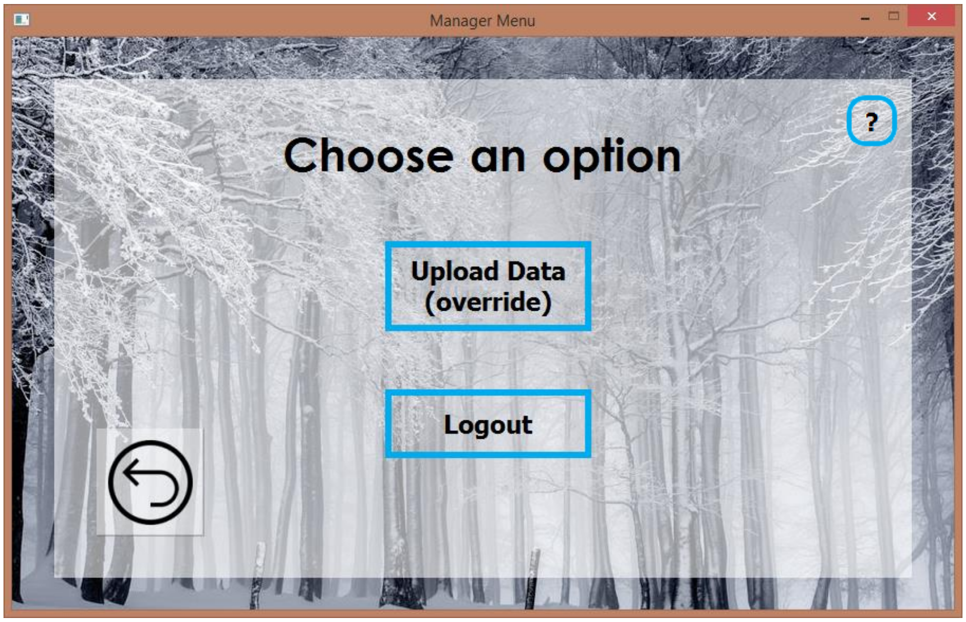


*Figure 16: **Screen 1.2.2** – the manager's menu screen. Clicking on "Back" will get the user to the home screen (**1**) without executing a logout action. Clicking on "Upload Data" will get the user to **screen 1.2.3.1**. Clicking on "Logout" will get the user to the home screen (**1**) after executing a logout action*

In both upload screens (**1.2.3.1** and **1.2.3.1**), the manager can upload the relevant files. The system will notify for successful and unsuccessful uploads.

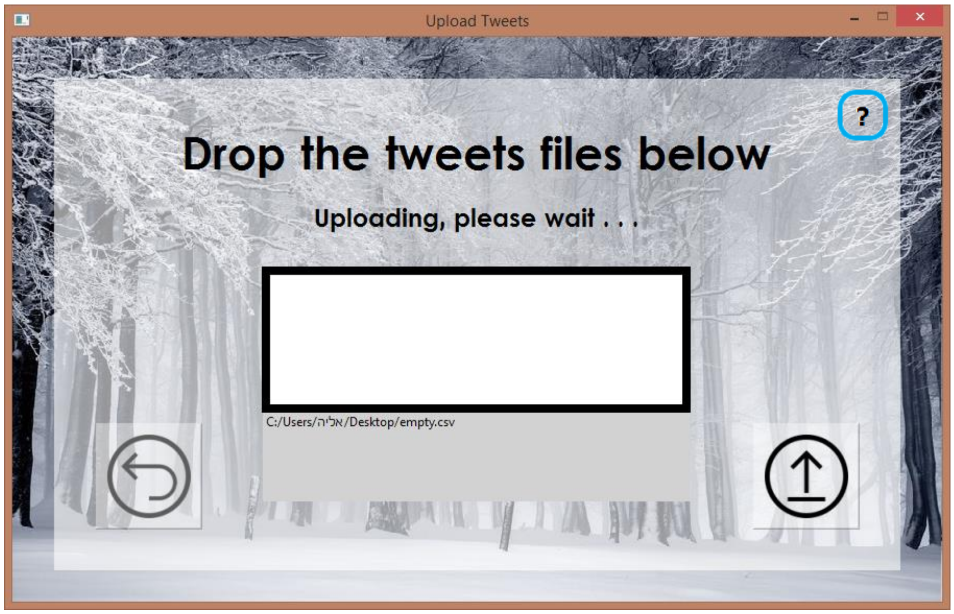


*Figure 17: **Screen 1.2.3.1** – the tweets upload screen. Clicking on "Back" will get the user to the manager's menu screen (**1.2.2**) without executing the upload action. Clicking on "Upload"(right button) will get the user to the scripts upload screen (**1.2.3.2**) after executing the upload action. The upload box will receive the uploaded files. In either case of successful or unsuccessful upload, the user will be notified*

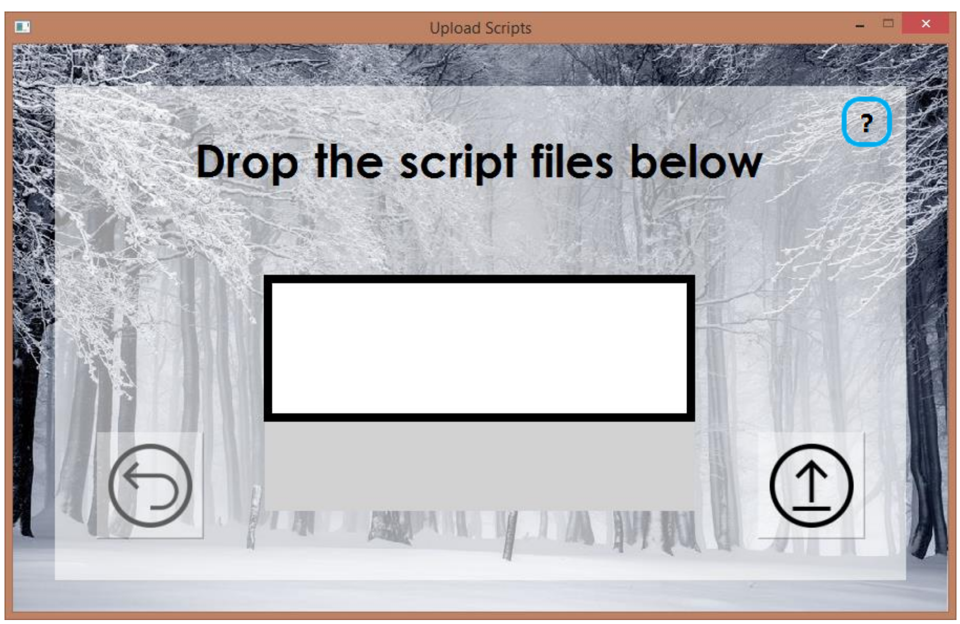


*Figure 18: **Screen 1.2.3.2** – the scripts upload screen. Clicking on "Back" will get the user to the the tweets upload screen (**1.2.3.1**) without executing the upload action. Clicking on "Upload" (right button) will get the user to the manager's menu screen (**1.2.2**) after executing the upload action. The upload box will receive the uploaded files. In either case of successful or unsuccessful upload, the user will be notified*

3.3 Testing

<u>Tweets and script lines analysis and scoring:</u>

Test ID	Description	Expected result	Actual result
ScoringAnger	Analysis of tweet that contains only one term, and it expresses anger only	The tweet's emotions scoring vector holds '1' for anger, '0' for other emotions	Pass
ScoringFear	Analysis of tweet that contains only one term, and it expresses fear only	The tweet's emotions scoring vector holds '1' for fear, '0' for other emotions	Pass
ScoringDisgust	Analysis of tweet that contains only one term, and it expresses disgust only	The tweet's emotions scoring vector holds '1' for disgust, '0' for other emotions	Pass
ScoringSadness	Analysis of tweet that contains only one term, and it expresses sadness only	The tweet's emotions scoring vector holds '1' for sadness, '0' for other emotions	Pass
ScoringJoy	Analysis of tweet that contains only one term, and it expresses anger only	The tweet's emotions scoring vector holds '1' for anger, '0' for other emotions	Pass
ScoringAnticipation	Analysis of tweet that contains only one term, and it expresses anger only	The tweet's emotions scoring vector holds '1' for anger, '0' for other emotions	Pass
ScoringSurprise	Analysis of tweet that contains only one term, and it expresses anger only	The tweet's emotions scoring vector holds '1' for anger, '0' for other emotions	Pass
ScoringTrust	Analysis of tweet that contains only one term, and it expresses anger only	The tweet's emotions scoring vector holds '1' for anger, '0' for other emotions	Pass
ScoringNegativeTerm	Analysis of tweet that contains a two-words term, in which the first is a negation word	The tweet's emotions scoring vector holds '1' for each emotion that opposes the negated word emotions, '0' for all the other emotions	Pass
ScoringAmbiguousNoContext	Analysis of tweet that contains only an emotionally ambiguous word, while the relevant episode holds no context for this word	The tweet's emotions scoring vector holds '0' for all the other emotions	Pass

ScoringAmbiguousContext	Analysis of tweet that contains only an emotionally ambiguous word, while the relevant episode holds a context for this word	The tweet's emotions scoring vector holds '1' for each emotion that is valued in the episode's context of this word, '0' for all the other emotions	Pass

<u>Tweets cleaning:</u>

Test ID	**Description**	**Expected result**	**Actual result**
TweetsCleaningForeignWords	Cleaning of tweet that contains two or more foreign words that appears in the database	Tweet gets deleted from the database	Pass
TweetsCleaningCharacterName	Cleaning of tweet that contains a name of a character that appears in the database	Clean form of the tweet contains the original name of the character	Pass
TweetsCleaningEmoji	Cleaning of tweet that contains a Unicode of an emoji that appears in the database	Clean form of the tweet contains the emoji's representative word as appears in the database	Pass
TweetsCleaningLematization	Cleaning of tweet that contains a word in an inflectional form	Clean form of the tweet contains the word in its original lingual form	Pass

<u>Script lines cleaning:</u>

Test ID	**Description**	**Expected result**	**Actual result**
ScriptLineCleaningCharacterName	Cleaning of script line that contains a name of a character that appears in the database	Clean form of the script line contains the original name of the character	Pass
ScriptLineCleaningLematization	Cleaning of script line that contains a word in an inflectional form	Clean form of the script line contains the word in its original lingual form	Pass

Upload of the files that contain the tweets:

Test ID	**Description**	**Expected result**	**Actual result**	**Comments**
UploadTweetsGoodCsv	Type of file is .CSV with expected number of fields	Upload Tweets succeeded	Pass	Shows message that states the number of uploaded tweets
UploadTweetsNonCsv	Type of file is different from .CSV	Upload Tweets denied	Pass	
UploadTweetsBadCsv	Type of file is .CSV with unexpected number of fields	Upload Tweets denied	Pass	Shows message that states the number of uploaded tweets (0)
UploadTweetsGoodBadCsv	Type of file is .CSV with rows that match the expected number of fields and others that don't	Upload Tweets succeeded partially	Pass	Shows message that states the number of uploaded tweets, equals the number of rows that match the expected number of fields

Upload of the files that contain the scripts:

Test ID	Description	Expected result	Actual result
UploadScriptsTxt	Type of file is .TXT, file name is a number	Upload Scripts succeeded	Pass
UploadScriptsNonTxt	Type of file different from .TXT	Upload Scripts denied	Pass
UploadScriptsTxtNonNumber	Type of file is .TXT, file name is not a number	Upload Scripts denied	Pass

Login:

Test ID	Description	Expected result	Actual result
LoginSucceeded	User inserts correct userID and correct password	Login succeeded	Pass
LoginIncorrectUserID	User inserts incorrect userID and correct password	Login denied	Pass
LoginIncorrectPassword	User inserts correct userID and incorrect password	Login denied	Pass
LoginIncorrectUserIDPassword	User inserts incorrect userID and Incorrect password	Login denied	Pass
LoginNoInput	User clicks on login without input	Login denied	Pass
LoginLogout	User logs in and clicks on Logout at the manager’s menu	Logout succeeded	Pass

4. RESULTS AND CONCLUSIONS

4.1 Results

The CSV files that were uploaded to the system, contained approximately three million tweets. While the upload process, few hundreds of tweets were omitted, due to commas that were used in the original text, in a way that interfered with the CSV structure. Yet more significant amount of data was lost in the emotion analysis phase. Since the system uses only tweets that has emotional value, it deleted the ones that - according to the algorithm - did not represent any emotion at all. In result, by the end of the emotion analysis level, about 1.05 million of the tweets remained in the database.

The scripts that were uploaded, described not only complex text, but also locations, actions and occurrences in short subjective notations. Thus, a large part of the scripts was found not to hold any emotional value. Around one thousand out of about two thousand and a four hundred script-lines were lost in the emotion analysis process.

The upload of the CSV files was programed to be done for whole file/s, using one of the database's commands for efficiency reasons. This command does not check for duplicates before inserting the data, so upon new upload, the system was ordered to delete the currently hosted data to avoid conflicts or duplications. Although the incontinence that might be aroused by that, it was found necessary, because checking for duplications would take several hours, while running the direct command only takes few seconds.

In each of experiments 1-8, we attempted to evaluate the performances of the system's emotion analysis procedure. At each experiment, we ran the emotion analysis procedure over pairs of tweets. We focused on one emotion at a time, so we chose pairs that significantly expressed the examined emotion. The tables below describe the input texts (the original tweets), alongside a corresponding normalized form (between 0-1) of their scored emotions vectors, as received from system's analysis process.

We expected the system to give the highest value (1) to the examined emotion, or for it to get at least the second-highest value. In the experiments, the examined emotion reached the highest value at 15/16 of the tweets, and the second-highest at the remaining one.

<u>Experiment 1 – Joy</u>

Input	**Score**							
	<u>Joy</u>	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
I'm so glad I finally found someone that loves Game of Thrones as much as I do	**1**	0.333	0	0.333	0	0	0	0.667
Game of thrones is really good I'm falling in love with Daenerys	**1**	0.5	0	0.5	0.5	0	0	0.5

Experiment 2 – Trust

Input	Score							
	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
Apparently, I look like a tan Daenerys Targaryen from game of thrones and I decided to fully embrace it	1	**1**	0	1	0	0	0	0.5
Lord Varys was a Daenerys supporter since day one	1	**1**	0	0.5	0	0	0	0.5

Experiment 3 - Fear

Input	Score							
	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
Daenerys is actually scary when she's talking about the war... ruthless dragon queen indeed	0	0	**1**	0	0	0.5	0.5	0
Every time I see a horse on game of thrones, I fear for its life I get so anxious	0	0	**1**	0	0.5	0.5	0.5	0

Experiment 4 – Surprise

Input	Score							
	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
I just watched game of thrones for the first time and wow that was something	1	0	0	**1**	0	0	0	0
Wow. Wow. Game of Thrones never Cersies to amaze me	1	0	0	**1**	0	0	0	0

Experiment 5 – Sadness

Input	Score							
	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
The fact that I missed the premiere of game of thrones last night makes me so sad	0	0	0	0	**1**	1	0	0
Game of Thrones already made me cry and I'm on the second episode	0	0	0	0	**1**	0	0	0

Experiment 6 – Disgust

Input	Score							
	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
wrote about Sam's disgusting poo and stew montage from Dragonstone. so gross.	0	0	0.5	0	0.5	**1**	0.5	0
I forgot how gross Game of Thrones can be #yuck	0	0	0	0	0	**1**	0	0

Experiment 7 – Anger

Input	Score							
	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
People love Jon snow so much they're blinded to how stupid this guy actually is. I'm so angry	0	0	0	0	0	1	**1**	0
They betrayed Jon Snow! I'm furious!	0	0	0.25	0.25	0.5	1	**1**	0

We noticed that other close emotions sometimes shared the highest value with the examined emotion. We focused on the currently examined emotion and did not consider it a problem, since the emotions are close.

Experiment 8 – Anticipation

Input	**Score**							
	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
I haven't been this excited over a TV show like Game of Thrones in a long time and can't wait to see what's next for season 7!	0.5	0	0	1	0	0	0	**1**
Winter is finally here! Can't wait for tonight's episode!	0.667	0.333	0.333	1	0.667	0.333	0.333	**0.667**

In each of experiments 9-10, we attempted to evaluate the performances of the system's linear-regression - based prediction. At each of these experiments, we ran the system's main consolidation process over one user's query. The system gathered the relevant tweets and received the result. Simultaneously, the system built eight linear regression models, one for each emotion, to predict the main process' result. We inserted a user query that returned at least 1,000 tweets. The linear regression models used 800 of them as training data, and the rest 200 as testing data.

The tables below describe the user queries that the system was given, alongside a corresponding normalized result of the system's main consolidation process, number of relevant gathered tweets it was produced from, the result of the system's prediction process and its minimum mean square error (one for each emotion).

We expected to get prediction that would approximate the system's consolidation results at accuracy level of at least 20%. We calculated the ratio between the results by the following formula for each emotion, where e represents the examined emotion (e.g. $\mathrm{Consolidation} - \mathrm{Prediction\ Ratio}_{\mathrm{Anger}}$).

$$\mathrm{Consolidation} - \mathrm{Prediction\ Ratio}_{\mathrm{e}} = \frac{P\mathrm{rediction}_{\mathrm{e}}}{Consolidation_e}$$

Hence, we required that the ratio would be between 0.8 to 1.2. In experiment 9 we got 7/8 of the ratios were in this range. In experiment 10 we got 6/8 of the ratios were in this range.

Experiment 9 – Episode: #2, Category: character, Character: Arya Stark

Results	Score							
	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
Original Consolidation	0.55	1	0.29	0.29	0.45	0.16	0.14	0.60
Prediction	0.46	1	0.20	0.27	0.37	0.13	0.13	0.60
Consolidation-Prediction Ratio	0.83	1	0.68	0.93	0.82	0.81	0.92	1
Minimum mean square error	0.08	0.08	0.03	0.04	0.07	0.03	0.04	0.07

Experiment 10 – Episode #5, Category: house, House: Targaryen

Results	Score							
	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
Original Consolidation	1	0.78	0.40	0.57	0.44	0.39	0.43	0.51
prediction	1	0.71	0.39	0.58	0.51	0.66	0.68	0.53
Consolidation-Prediction Ratio	1	0.91	0.97	1.01	1.15	1.69	1.58	1.03
Minimum mean square error	0.36	0.35	0.08	0.07	0.07	0.08	0.09	0.37

In each of experiments 11-12, we attempted to evaluate the emotional value difference between the two type of data that we checked. We ran the system's main consolidation process over one user's query. The system gathered the relevant tweets and script-lines and received two normalized emotions vector as result.

The tables below describe the user queries that the system was given, alongside a corresponding normalized result of the system's main consolidation process over the relevant script-lines in one line and the relevant tweets in another line. The table also contains the number of relevant gathered tweets and script-lines the results were produced from.

We expected that the system's consolidation results over the tweets and script-lines would be close at accuracy level of at least 25%. We calculated the ratio between the results by the following formula for each emotion, where e represents the examined emotion (e.g. $\text{Tweets} - \text{Scripts Ratio}_{\text{Anger}}$).

$$\text{Tweets} - \text{Scripts Ratio}_{\text{e}} = \frac{Tweets_{\text{e}}}{Scripts_{e}}$$

Hence, we required that the ratio would be between 0.75 to 1.25. In experiment 11 we got 4/8 of the ratios were in this range. In experiment 12 we got 4/8 of the ratios were in this range.

Experiment 11 – Episode: #2, Category: character, Character: Arya Stark

Results		**Score**							
	amount	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
Tweets	1737	0.55	1	0.29	0.29	0.45	0.16	0.14	0.60
Script-lines	37	0.75	1	0.37	0.81	0.56	0.50	0.43	0.50
Tweets-Scripts Ratio	46.94	0.73	1	0.78	0.35	0.80	0.32	0.32	1.2

Experiment 12 – Episode #5, Category: house, House: Targaryen

Results		**Score**							
	amount	Joy	Trust	Fear	Surprise	Sadness	Disgust	Anger	Anticipation
Tweets	6741	1	0.78	0.40	0.57	0.44	0.39	0.43	0.51
Script-lines	53	1	0.89	0.65	0.74	0.96	0.78	0.74	0.50
Tweets-Scripts Ratio	127.18	1	0.87	0.61	0.77	0.45	0.5	0.58	1.02

4.2 Challenges and Limitations

We decided to delete tweets whose language is not English, because we did not implement translation in the system. In retrospect, we could translate these tweets into English with existing libraries that we discovered after a thorough investigation.

The system cannot identify cynical comments, which can change the intent of the sentence, as well as the emotions. In order to identify cynical comments or other context related terms, we suggest using deep learning techniques.

4.3 Conclusions

The emotion analysis itself was sometimes not satisfying; Upon several experiments, words and expressions that were meant to express certain emotions, were not scored correctly by the system, or sometimes they were not given any score at all. Yet, in most of the checked cases, the analysis results were close to be agreeable, or even reflected the exact expected emotions. It is important to add that the system analyzed correctly the majority of the two-words expressions it was given, mostly if they held a relation of negativity.

After experiments with some tweets and script-lines that were given wrong scoring, we could indicate that the lexicon that gave the words their emotional value was noticeably not accurate. In other cases, it entirely lacked groups of important words. That said, even after the lemmatization process worked very well on most of the words we checked. We concluded that using a lexicon that is both more accurate and more expanded, might optimize the analysis' quality. Doing so, we expect, will also reduce loss of useless tweets and script-lines, that were considered irrelevant.

The comparison of the tweets and the script-lines analysis results proven that there is but slight connection between the two. Since in our experiment we received very few script-lines compared to many tweets, we assume that perhaps if more script-lines were given, we could establish a definite connection between them.

References

[1] “Statista: the statistics portal”, 2018. [Online]. Available: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[2] R. Plutchik, “A psychoevolutionary theory of emotions”, *Social Science Information (SAGE, London and Beverly* Hills) 21: 529-553, 1982

[3] C. Darwin, “The expression of the emotions in man and animals”, *London: J. Murray*, 1872.

[4] P. Ekman & W. V. Friesen, “Constants across cultures in the face and emotion”, *Journal of Personality and Social Psychology, 17*(2), 124-129, 1971.

[5] W. Wundt, “Grundriss der Psychologie” (translate: “Outline of Psychology”), 1896. Available: https://archive.org/details/cu31924014474534/page/n4

[6] H. Schlosberg, "Three Dimensions of Emotions". Psychological Review. 61 (2): 81–88, 1954.

[7] S. Mohammad, “From Once Upon a Time to Happily Ever After: Tracking Emotions in Novels and Fairy Tales”, *Proceedings of the ACL Workshop on Language Technology for Cultural Heritage*, *Social Sciences, and Humanities (LaTeCH)*: 105-114, 2011

[8] K. Roberts, M. A. Roach, J. Johnson, J. Guthrie and S. M. Harabagiu, “EmpaTweet: Annotating and Detecting Emotions on Twitter”, Conference on Language Resources and Evaluation (*LREC)*: 3806-3813, 2012

[9] R. Kempter, V. Sintsova, C. Musat and P. Pu, “EmotionWatch: Visualizing Fine-Grained Emotions in Event-Related Tweets”, *Eighth International AAAI Conference on Weblogs and Social Media*: 236-245, 2014

[10] P. F. Brown, P. V. deSouza, R. l. Mercer, V. J. Della Pietra and J. C. Lai, Class-Based N-gram Models of Natural Language, Computational Linguistics 18 (4): 467-479, 1992

[11] “NRC Word-Emotion Association Lexicon (EmoLex)”, 2013. [Online]. Available: http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm