Effectively Classifying Texts Affectively:

An Analysis of the Combination of Lexical and Emotional Features in Text Classification

Master Thesis for the MA Digital Humanities

In partial fulfilment of the requirements for the degree of

Master of Arts

Anne Geer van Dalfsen

University of Groningen

August 2018

Supervisor: dr. M. Nissim

Second reader: dr. F. Harbers

Table of Contents

Abstract

The issue of not being able to accurately classify items is present throughout various media. Recommendations are an example of this. As they are often based solely on arbitrary and malleable genres, recommendations can be wildly inaccurate. For the medium of text, the presents study proposes a text classification approach that combines both lexical and emotional features. This combination should comprehensively represent the most important elements of a text. This in turn should allow for detailed comparisons and thus accurate recommendations. In order to explore this possibility, the present study classifies the pre-categorised texts contained in the Brown Corpus based on a combination of lexical and emotional features. The classifier and additional statistical tests produced numerous results. Even though there were various limitations, the results suggest that a combination of lexical and emotional features is indeed more effective than either being used separately.

Keywords: digital humanities, text classification, lexical features, emotional features

1 – Introduction

In my personal life, I am an avid reader of fantasy fiction. Whilst purchasing or keeping track of books online, websites often recommend other, related items. However, these recommendations vary wildly in usefulness, as at times the recommended book is far from as enjoyable as the one the recommendation was based on. Although books might be tagged to be of the same genre, their contents do not have to be anything alike. In truth, this is not very surprising. Although the books may have the same tagged genre and even similar keywords in the blurb, the style of writing can be completely different. Furthermore, a single genre can be incredibly broad. For instance, there are various subgenres of fantasy, such as high, low, urban, and steampunk. In my experience, however, most websites will simply file them all under fantasy. Moreover, even when using the subgenres, a book's exact genre is not always clear, causing fans to quibble over what genre it is exactly. This begs the question of what genre actually is and what causes such issues.

Genre, much like topic and keywords, can be used to infer the content of a text and subsequently use that to categorise said text. By equating it to a textual feature such as topic, genre seems to be a fairly straightforward concept. However, establishing what it entails exactly in an academic setting is somewhat more challenging. This is largely due to the many different definitions that can be found in scientific literature. Biber (1988), for instance, uses the "term 'genre' to refer to text categorizations made on the basis of external criteria relating to author/speaker purpose" (p. 68). Biber explains that genres such as "'Press reports' are directed toward a more general audience than 'Academic prose'; the former involves considerable effort toward maintaining a relationship with its audience, and is concerned with external temporal and physical situations in addition to abstract info" (1986, p. 390-391). Each genre thus has a specific

purpose in communicating with its audience. As such, because it has a specific communicative purpose, a pamphlet is considered to be as genre just as much as a news article might be. Furthermore, although phrased somewhat differently, others agree with Biber. Hyland states that genre analysis requires two assumptions, namely that "the features of a similar group of texts depend on the social context of their creation and use" and that "those features can be described in a way that relates a text to other texts like it" (2002, p. 114). Swales further corroborates this by saying that "genre comprises a class of communicative events, the members of which share some set of communicative purposes" (2008, p. 58). One addition is made by Hyland (2002), as he mentions that other external constraints can also define a genre. Swales (2016) explains this by referring to the changes made to the genre of academic writing. Initially such texts contained solely an academic report. Eventually, however, publishers required authors of such articles to add an abstract, which was later followed by key words, highlights, etc. However, even though this overarching definition of genres being defined by external criteria appears feasible, it might make more sense to categorise texts based not solely on external factors.

The opposite of external factors would be the factors contained in the actual texts themselves. As such, genres could also be defined based on commonalities of structural and linguistic features found in texts; internal criteria (Kessler, Nunberg & Schütze, 1997; Karlgren, 1999). Unfortunately, while this makes sense at first glance, it is a fundamentally flawed approach. Constraining genres by the language used would result in them being too narrowly and rigidly defined. This would in turn inevitably become problematic, as genres are inherently fluid concepts (De Geest & Van Gorp, 1999). Indeed, genres have been shown to evolve over time. Haan-Vis & Spooren (2016) found that the language used in Dutch journalistic subgenres became vastly more informal over the course of 50 years. As such, defining genres with too strict

requirements would eventually cause the genre structures to become incomplete and flawed. Biber (1988) provides a solution to this issue by arguing that external criteria and internal criteria ought to be seen as independent; genre and text type, respectively. In doing so, genre becomes defined by external criteria such as its communicative purpose, target audience, and constraints. However, this definition allows genres to contain a considerable amount of variation. Consider the genre 'news', for instance. It has the communicative purpose to provide its audience with an accounting of events, which allows its texts to have a wide range of topics. Conversely, genres that have a less broad purpose also have a more narrow range of topics. An example of this latter type can be found in grocery receipts (Swales, 2016). The issue that arises with this is that it can become challenging to determine the exact genre of a text.

A text that adheres to a genre's communicative purpose and other external constraints is considered to be prototypical of that genre (Swales, 2008; 2016). However, even if a text adheres to those requirements to a lesser extent, it can still be identified as belonging to that genre. The problem that this creates is that there is no clear, indicative boundary that separates genres. 'Generic integrity', as Bhatia calls it, " is not something which is static or 'given', but something which is often contestable, negotiable and developing, depending upon the communicative objectives, nature of participation, and expected or anticipated outcome of the generic event" (2004, p. xi). The solution most commonly used is a system of categories and subcategories. Once two texts differ enough from each other to not consider them to be the same, they can form their own subgenre, while still falling under the same overarching genre. However, this system has its own flaws (De Geest & Van Gorp, 1999). The main issue is that genres can be split up into increasingly smaller subgenres to an extent that is neither feasible nor useful. However,

there is no clear indication that this should not be done. At the same time, the broader overarching genres are not particularly useful when providing recommendations, either.

In other words, a definition of genre based on external criteria is useful and roughly indicative, but too fluid and inexact. Moreover, a definition that includes internal criteria is too narrow and rigid. Finally, relying on a system that utilises a structure of increasingly more niche subgenres to deal with the issue is problematic as well. Summarised, recommendations based solely on genre are limited at best. More detailed information about the content of the text itself is required to accurately find texts that are similar in more than just the communicative purpose. The most obvious solution is a combination of genre and such information. However, such data is generally not readily available for all books. As such, I would like to propose an objective and quantified approach to this matter.

In an essay on analysing films using digital methods and means, Hoffman, Brouwer, and Van Dalfsen (2018) analysed such methods. One method involved taking a still image of the film at every second and subsequently superimposed all of these images using the software ImageJ (*Schneider, Rasband, & Eliceiri, 2012*). This would allow for a general indication of colour tones, camera angles, etc. Another method involved extracting the audio frequency data from the film, again sampled at every second. This data could then be used to draw a graph that showed the average frequency of the audio at that point in the film. As a number of studies have shown, such frequency data can be used to determine the emotion in music and acted speech by looking at the key (e.g. minor and major; sad and happy) in which the sound is set (Schreuder, Eerten, & Gilbers, 2006; Kamien, 2008; Gilbers et al. 2010). There are also a number of other methods that provide information about, for example, scene length, and speed of camera movements (Heras, 2012; Ferguson, 2017). Hoffman et al. (2018) theorised that using the information that these

methods provide could tremendously improve film recommendations. These methods would enable them to base their recommendations not only on broad, generalising terms, but on more detailed, quantified values that are inherent to films. A similar type of approach can be used to do the same for texts.

Text classification finds its origin in the 1960s and has been actively studied since the 1980s. In the late 1990s it was combined with machine learning techniques, becoming akin to how we know and use it today (Sasaki, 2009). As opposed to doing so manually or creating automatic classifying rules by hand, most modern-day approaches to text classification are based on statistics. Using machine learning methods in combination with human-annotated training data, a machine can automatically learn the most useful classification rules with which to identify the class (e.g. genre) of a text (Joachims, 1999; Yang & Joachims, 2008). Although it can quite quickly become incredibly complex, text classification will always be a matter of classes and their features at its core. For instance, if the texts in Class A have high values for a certain feature, while the texts in Class B have low values for the same feature, any texts with a high value will be classified as Class A, and those with low values as Class B. Those features can be any quantifiable element of a text: how many times a certain word is used, general word count, unique token count, how many upper case letters are contained in the text, etc. Because of this inherent versatility, it is an ideal instrument to determine which text another text is most alike. In other words, it could be used to make accurate, objective recommendations.

The main issue is in this approach is that a text has numerous elements, and thus features, that make it that particular text. For instance, using just lexical features to compare texts would indicate whether the language use from a technical point of view was similar or not. However, doing so would ignore any affective language, which is integral to the tone and feel of the text.

Conversely, using just emotional features would ignore a part of language use as well. Therefore, logically speaking, both lexical and emotional textual features ought to be used to compare texts properly. Both types of features have been studied and analysed in fairly recent years. However, insofar as can be determined, the two feature types have not been studied in conjunction with one another.

As a result, the primary focus of this thesis is to determine the value of combining lexical and emotional textual features for the purposes of text classification. Secondary aims include determining the efficacy of the dataset and both of the emotional lexicons that will be introduced in the next section.

Before getting to the thesis proper, I would like to mention that a considerable amount of the work spent on this thesis has gone into the creation of the programs used to collect the data and perform the classifications. At times, these programs will be referred to, which will have the following structure: "(ProgramName.py - lines ## and/through ##)". The reason for this is because, when combined, both programs reach a length of roughly 2000 lines, making them rather unwieldy to present them in their entirety in the thesis itself. Snippets of code will be present in the thesis with the same reference should you wish to look at the context. Much the same is the case for the results. As such, the main results file will also be referenced to at times. The structure that this reference will have is: "(Results.xlsx – Sheet: 'SheetName'). For this reason, the programs, the lists of data they use, as well as all the results of the study have been hosted in a GitHub repository where they can be perused at your leisure. The folders should be structured in a straightforward manner and the files should have self-explanatory names.

GitHub repository: https://github.com/AvDalfsen/Master-Thesis.

2 – Background

According to Zechner (2013), text classification while employing machine learning is based on a total of five elements. The first element is the type, which concerns whether or not the classification process is supervised. Supervised classification entails the training of the classifier in the desired target properties and classes, whereas unsupervised classification leaves the classifier to independently identify the properties and classes. Each type has its uses, but thus far most work has focused former. The second element is the target, which relates directly to the purpose of the study; it concerns the goal of the classification process. The third element is the corpus, which, in the case of text classification, is the collection of texts that will be used to test the classifier. The corpus used is important as the size (e.g. number of texts) will have an effect on the performance of the classifier. Too few texts will result in an insufficiently trained classifier and unclear differences between classes. A corpus can also contain additional data or metadata, such as whether the author is male or female, what year it was published, etc. Such metadata is useful for studies that intend to use that information to, for example, determine whether a link exists between author and topic. In short, an incorrect choice in corpus can severely limit the performance of the classifier, as well as the range of possible targets. The fourth element is the features. Features are used to distinguish classes from one another. For instance, if the texts belonging to one class have a consistently higher word count than another class, then the word count feature serves as a way to identify the class of a given text. The most common feature category includes lexical features, such as "average lengths of words, sentences, paragraphs or texts, as well as a few complexity measures, and perhaps most importantly word frequencies" (Zechner, 2013, p. 1). Somewhat less common, though they have been gaining more attention in recent years, are emotional features, which can include any emotion so long as

data about said emotion is available and usable. Regardless of the features used, prior to testing it is most often impossible to be entirely certain whether a specific feature will be useful as a way to distinguish one class from another. In fact, it is exactly because of this that Koppel, Argamon, and Shimoni (2002) purposefully started with a total number of 1081 features and through a process of elimination they determined what features were useful, which they then used for the actual study. The fifth element is the classifier. The type of classifier used (e.g. decision trees, support vector machines, neural networks, Bayesian classifiers, etc.) will also have an effect on the results. However, this effect can generally also not be determined before running the actual test (Petrenz, 2009). All five elements are important to the efficacy of a classifier, but the present study intends to focus on the fourth element: features.

What the classification process starts with is a feature set and a number of classes. The feature set can essentially be any quantifiable element of a text. Classes are similarly broad. Genre, for example, is simply a type of class; fiction is one class and non-fiction is another. Each class has a different value for each of the features. Therefore, by training the classifier with samples that each have an assigned class as well as a value per each feature, it will be able to analyse a text, extract the values for each of the features, and determine which class it is most likely to be (Yang & Joachims, 2008). So long as the classes, feature set, and the values for each feature per class are known, a classifier can be trained to classify texts based on those features. As such, it has numerous applications.

One of the first and better known applications of text classification was a binary distinction between two classes: desired texts and undesired texts. By separating the two, there was no, or at least far less of an issue with combing through messages and finding the ones that are actually interesting and useful. The result is what we know today as spam filters (Joachims,

1999; Androutsopoulos, 2000). Another, albeit less straightforward example is the study of Koppel et al. (2002), which attempted to determine whether or not the gender of the author could be determined by looking at "simple lexical and syntactic features" (p. 1). The lexical features they used included the word count, token count, lexical richness, pronouns, specific determiners, the other determiners, negations (not/*n't), the preposition 'of', and the remaining prepositions. They employed machine learning algorithms on texts taken from the British National Corpus, which were tagged for gender and genre. As they analysed their findings, they realised that the results indeed showed whether the author was male or female with an accuracy of approximately 80%, but they could also be used to distinguish fiction from non-fiction with an accuracy of 98%. These positive and tremendously accurate results suggest that those features should prove equally useful in similarly distinguishing more specific genres from one another. As such, the present study will include a number of the features used in Koppel et al.'s study (2002). Due to the limited allotted time for the present study, however, they will be limited to the first three. Moreover, these features will be combined with those from the fairly common and standard Bag-of-words approach (Yang & Joachims, 2008). Although both sets of features will later be explained in greater detail, suffice it to say for now that together they will form the lexical features for the present study. Since the goal of this study is to determine the efficacy of a classification method based on both lexical and emotional features, the emotional feature set also needs to be established.

The main issue with attempting to set up an experiment to test the usefulness of classifying texts based on affective language is that the emotional valence that words carry depends entirely on the framework of the producer and the receiver of the words. The reason for this is because "language helps constitute emotion by representing conceptual knowledge"

(Brooks et al., 2017, p. 180). In fact, “learning to label feelings is at the core of many types of psychotherapy” as it “helps a person regulate their feelings” (p. 180). Therefore, if a certain word is associated with pleasant memories, it will evoke those pleasant memories, resulting in a similarly pleasant perception of the word. Furthermore, it is entirely possible to “re-conceptuali[se] the meaning of a feeling with a different linguistic category … [to] help regulate emotions by helping transform one type of experience (e.g. fear) into another (e.g. anger)” (p. 180). This suggests that emotions evoked by a certain word can differ depending on the receiver. Because of this, to get a proper indication of which emotions are evoked by which words, input from multiple participants is required.

Once such a resource has been completed or acquired, however, interesting avenues can be explored. One of the most common applications that are based on affective language is determining the sentiment contained in a text. Semantic analysis has been well researched in the past few years due to the readily available data on Twitter, product reviews, and other such resources (Sarlan, Nadam, & Basri, 2014; Joyce & Deng, 2017; Srivastava, Singh, & Kumar, 2014). With regard to the range of emotions, however, most of those types of applications tend to classify texts into up to a maximum of three emotions: negative, neutral, and positive. Because of the target of the studies, those emotions are exactly sufficient. For studies involving other emotions, however, they would not be. When attempting to classify texts into genres, for instance, one would need more than those three emotions. Because emotions are expressed differently in different genres (Ofoghi & Verspoor, 2017), there is a wide range of emotions that can be useful in text classification. Samothrakis and Fasli’s study (2015), for instance, showed that the “six basic emotions” (p. 1) (anger, disgust, fear, joy, sadness, and surprise) can be used to classify texts into genres (science fiction, horror, western, fantasy, crime fiction, mystery,

humour, and romance) with an accuracy significantly higher than random chance (varying between 0.42 and 0.58 (with a maximum of 1), depending on settings and classifier used). Furthermore, the study showed that the emotion of fear was "the most important differentiator between genre novels" (p. 1). This suggests that certain emotions are more useful than others for text classification. However, the usefulness will likely depend on the dataset used and should therefore not be assumed prior to testing. In order to thus ensure a wide enough range of emotions, while still remaining feasible, the present study will be employing two emotional lexicons. The first of which is the Dictionary of Affect in Language.

In 1989, Whissell published the first edition of the Dictionary of Affect in Language (henceforth: Dictionary). The Dictionary consisted of over 4000 English words, each of which came with a score ranging from 1 to 3 in two dimensions: Evaluation and Activation. The data upon which the Dictionary was based were obtained by input from 73 participants. The Dictionary was used in a number of studies that dealt with the memorisation of words and eventually also to analyse the emotional levels in literature. Ultimately, it became apparent that the Dictionary was too limited in its potential uses due to the fact that it had a far too low matching rate when used to analyse literature. Realising that a tool that "quantifie[d] the emotional undertones of natural language would be useful in a variety of settings" (2009, p. 509), Whissell undertook a revision of the Dictionary. This revised version is the one that the present study will be employing.

The revised version of the Dictionary more than doubled the number of words that could be found in the old version, as it contained close to 9000 words. The selection process for words in the original version focused mainly on words that were emotionally laden, thus resulting in clearly defined scores. However, this turned out to be the limiting factor with regard to the

matching rate. Because of this, the word selection for the revised version "was designed to privilege natural language" (Whissell, 2009, p. 510). As a result, the majority of the words, over 75%, were chosen based on their frequencies in the 1967 edition of the Brown Corpus (Francis & Kuçera). All words with a frequency higher than 10 million that appeared in two or more texts from the corpus were included. The resulting list went through a process of elimination until Whissell ended up with a list of 8,742 words. This list was then tested using 16 100-word samples of several types of texts, from newspapers to song lyrics. Where the old Dictionary had a matching rate of 19%, the revised version had one of 90%.

Furthermore, contrary to the two dimensional scores of the old Dictionary (Evaluation and Activation), the revised version had three dimensions: Pleasantness and Activation, which "are the two chief dimensions of affective space" (Whissell, 2009, p. 510), and Imagery, which was defined as the ease with which people could "form a mental picture" of the word. "Imagery plays a role in learning and memory … but it is also important in natural language where it serves as an indicator of abstraction" (p. 510). Now confident that the Dictionary contained the necessary words and ratings to properly analyse natural language, Whissell attempted to determine how genres differed with regard to the three dimensions.

The texts that Whissell analysed using the revised Dictionary were plays by Shakespeare (2007). More specifically, she wished to determine whether there was a quantitative difference between the comedy and tragedy genres by looking at the affective language contained in the plays. After using a program to score over one million words, her findings indicated that comedy plays used significantly more words with a high Pleasantness rating than tragedy plays. Tragedy plays, however, employed more words with a high Activation rating. Subsequently, Whissell created a "discriminant function formula based on Pleasantness, Activation, and Imagery [that]

was able to identify genre with high accuracy" (Whissell, 2007, p. 189). Her findings resulted in a positive answer, as the results indicated that the Dictionary could certainly be used to classify the selected plays into those two genres. However, Whissell did note one limitation of the results. She noted that they were "an impoverished source of information. The numbers cannot stand in lieu of the plays because the plays included many pleasing complexities of meaning, unending adventures in vocabulary, and a human element completely absent from the numbers" (p. 190). She certainly has a point in that the results do not shed light on the content of the texts analysed, particularly with regard to their cultural significance and use of language. However, the three dimensions should more than suffice to aid in answering the question of the present study.

However, using solely this approach for the emotional feature set would ultimately likely end in failure when applying it to more than two classes. Whissell's study (2007) focused on the differences between two genres that most people would deem directly oppositional; comedy and tragedy. Because of this, though based on nothing more than a presumption at this time, a classification approach based solely on the Dictionary would likely encounter issues when classifying genres that are too similar in these three aspects. Therefore, a broader approach should be included in order solve this issue by providing more emotional features to aid in differentiating classes from one another. To this end, the present study will also be employing the features contained in the EmoLex.

As mentioned, to properly study the emotions in texts, input from multiple participants is required. This naturally constrained any "research in emotion analysis [as it] had to rely on limited and small emotion lexicons" (Mohammad & Turney, 2013, p. 1). This limitation encouraged Mohammad and Turney (2013) to create the Emotion Lexicon; the EmoLex. The EmoLex consists of words that are labelled based on eight emotions: anger, anticipation, disgust,

fear, joy, sadness, surprise, and trust. Negative and positive sentiments were included as well. The annotations for the lexicon were done via crowdsourcing. By employing crowdsourcing, the words were annotated manually by Mechanical Turkers. The Turkers had to answer questions, which they in return received money for. The Turkers that could participate were limited solely by whether they were either native or fluent speakers of English. In order to ensure the responses they paid for were actually useful, each Turk had to answer certain questions for which there was a gold standard. By doing so, any responses by Turkers that did not meet the gold standard were ignored. Moreover, any responses that did not follow the instructions (e.g. 'answer all questions before moving on') were ignored as well. The end result of the pilot version of the lexicon contained 2000 words that were accurately annotated according to emotions (Mohammad & Turney, 2010). Encouraged by their success, Mohammad and Turney (2013) then moved on with a similar approach to enlarge their lexicon. This lexicon ultimately ended up containing a total of 14,182 annotated words. Although no exact number is mentioned, considering that the EmoLex is even larger than the Dictionary, it is presumed to have a similarly high matching rate as the Dictionary. As such, it should serve well in broadening the Dictionary's fairly limited range of emotions. Furthermore, it has shown several times during various classification tasks that its performance is up to par (Kiritchenko et al., 2014; Mohammad, 2012; Rosenthal et al., 2015).

In short, the present study will be employing two sets of lexical features and two sets of emotional features in a classification experiment. The goal of the experiment is to determine whether or not a combination of the two types of features yields better results than when using either separately.

3 – Method

3.1 – Procedure

The first step of the project was to find a dataset suitable for the goals of this study. Once this had been done, it had to be ensured that all texts adhere to the same structure and would be usable in Python. Once this had been confirmed, work started on the actual writing of the programs. The work would be done in its entirety in Python (version 3.6.4) using the program Spyder (version 3.2.7). Spyder is an open source interactive development environment for Python that is included in the similarly open source Anaconda distribution. When the first program had been completed, it would be used to collect the required data. Once that data had been compiled and processed, it would be used together with the second program to run the classifier. The results that this second program produced were the data upon which the remainder of the study would be based.

3.2 – Dataset

The dataset that this thesis used was the Brown Corpus (Francis & Kucera, 1979). The Brown Corpus consisted of 500 samples of texts, and had a total of just over one million words. Each sample consisted of about 2000 running words. Each of the texts, insofar as could be determined by the compilers, first appeared in print in 1961 and were all written by American English authors. “Verse was not included on the ground that it presents special linguistic problems different from those of prose” (Francis & Kucera, 1979). As such, all of the 500 samples were prose, though any quoted verse was still included. Furthermore, drama was excluded on the basis of it being an imaginative recreation of spoken discourse. For similar reasons, though various genres of fiction were included, samples that consisted of dialogue for

50% or more were excluded. Furthermore, although there were multiple versions of the corpus available, including one that had been entirely manually tagged for word class, the present study used the base version that only contained the original texts themselves. The reason for this was because that version was readily available in Python's NLTK (Natural Language Toolkit) package.

One of the more important features of this corpus for this particular endeavour was that all the 500 samples were categorised into genres and subgenres. These genres fell into one of two overarching categories: informative prose and imaginative prose; nonfiction and fiction. However, aside from the fact that the "list of main categories and their subdivisions was drawn up at a conference held at Brown University in February 1963" (Francis & Kucera, 1979), nothing was known about what definition of genre the compilers of the corpus adhered to. Unfortunately, because so little was known about the process that went into deciding the genres and subgenres, it was hard to say what effect it would have on the results. Table 1 below shows the genres contained in the Brown Corpus. One of these genres was called 'Miscellaneous', which, by its very definition, was a collection of various types of texts. It was thus unknown in what way the texts contained in that genre were related to one another. This was an issue because if the texts did not share lexical or emotional similarities, the classification performance for that genre would be poor. Though this issue was most obvious for the 'Miscellaneous' genre, the same was the case for all the other genres. The texts in a genre had to somehow have been related to each other for them be categorised as they were, but because the definition of genre used to do so was unknown, it was impossible to determine what this relation was. Even so, the fact that all texts were all divided into pre-existing categories meant that the texts could be used as the samples and the genres as the classes during the classification process. Therefore, the issue

of unknown relations between the texts of a given genre was deemed a variable that would have to be considered when analysing the results. Finally, each genre had at least two, but up to seven subgenres. For instance, the 'Learned' genre in nonfiction had the subgenres: natural sciences, medicine, mathematics, social and behavioural sciences, political sciences, humanities, and technology and engineering.

One downside to the corpus was that it did not have an equal number of texts for each genre. Indeed, as shown in Table 1, the number of texts per genre could differ quite a bit.

Table 1 – The genres contained in the Brown Corpus and the number of samples per genre

Genre	**Number of Texts**
Non-Fiction	
Press: Reportage	44
Press: Editorial	27
Press: Review	17
Religion	17
Skills and Hobbies	36
Popular Lore	48
Belles Lettres, Biography, Memoirs, etc.	75
Miscellaneous	30
Learned	80
Fiction	
General Fiction	29
Mystery and Detective Fiction	24
Science Fiction	6
Adventure and Western Fiction	29
Romance and Love Story	29
Humour	9

As such, although the subgenres provided a more detailed idea of what specific genre a text was, there were simply too few samples in the corpus to make that approach feasible. Using the subgenres as classes would thus likely have had a detrimental effect on the performance of the

classifier. As such, the 15 overarching genres would be used as classes in the present study. However, the sample size would remain to be an issue. The exact effects it will have on the results of the present study were unknown, but it was a variable that was kept in mind. As such, it will be addressed in the Discussion section.

Another reason for picking the Brown Corpus was because it was freely accessible either by simply downloading the text files or by accessing them as lists of tokens through the NLTK package in Python. As Python would be used to write the data collection program, using the Brown Corpus was very convenient.

Finally, as mentioned, over 75% of the words contained in Whissell's Dictionary were chosen based on the most frequently used words in the Brown Corpus. Using it as the database for the study should therefore have ensured that the matching rate would be as high as possible.

3.3 – Pre-processing

The idea behind the program was to create an interactive tool that would allow any user to obtain various types of data from either individual texts from the Brown Corpus, entire genres, every single text in the corpus, or a text provided by the user. To ensure that all texts were processed in the exact same way, any pre-processing of the text(s) was coded into the program. Seeing as the goal was to ultimately classify the texts based on their features, they were pre-processed in a number of ways. Pre-processing in this case did not include tokenisation, as the texts from the Brown Corpus, when accessing them via the NLTK package, had already been tokenised. The tokeniser used was, presumably, the 'WordPunctTokenizer', as all words and punctuation were separate tokens. However, other steps of pre-processing were taken.

Firstly, the decision was made to remove punctuation, even though the ultimate goal of the thesis did not necessarily require it. The question of the present study only requires a comparison of the validity of the combination of several approaches, not to create an approach with the highest accuracy possible. However, due to the tokenisation process, all forms of punctuation would be seen by the program as a word. This meant that, for every form of punctuation, the program would have to check whether or not it was in any of the lists (more on this in the next section). Due to the features used, leaving the punctuation in the texts would add nothing of value to the data. However, it leaving it in would result in a considerably longer time required for the program to process all texts. Because of this, all forms of punctuation were removed during pre-processing. The process by which this was done was to create a variable that contained all forms of punctuation encountered in the texts. This variable was then used by the program to go through each text, creating a new list containing all the words from the text that were not in said variable. This thus resulted in a list containing all the words from the text excluding any and all punctuation.

Secondly, all words in the texts were lowercased. This was a fairly crucial step. The reason for this is because the words in the lists of the Dictionary and the EmoLex were all lowercase as well. Python sees a word in lower case as different and separate from the exact same word in upper case or with even a single upper case letter. In other words, words with an uppercase letter would not find a match in any of the lists of the emotion lexicons. Moreover, the lowercase and uppercase versions would both be seen as unique tokens, which would skew those results.

Thirdly and finally, all ‘stop words’ were removed from the texts. The term ‘stop words’ generally refers to words that are represented too frequently in a text while not contributing any

real meaning that would distinguish one text from another. Lists of stop words usually include most, if not all articles, conjunctions, various frequently used verbs, etc. For simplicity and efficiency's sakes, NLTK's list of English stop words was used. The entire list can be viewed here: https://gist.github.com/sebleier/554280 (Retrieved on the 15th of July, 2017).

3.4 – Features

After the code for the pre-processing part had been written, focus shifted to the section of the program that pertained to the collection of the data. This section can be broken down into roughly four distinct parts.

3.4.1 – Lexical Features

The first part is the collection of general information about the texts, such as word count, unique token count, and lexical richness. This part was the easiest to accomplish as Python has functions perfectly suited for this purpose, namely: *len()* and *set().* The first of these functions provides the length of a string in characters or the number of items in a list. As the texts were in the form of a list, *len(text)* provides the word count. *Set()* essentially creates a set of each unique item. *'set(['1','2','1'])* would return: *{'1','2'}*; it prints the unique tokens in a text. Therefore, a combination of the two, *len(set())*, provides the number of unique tokens. An example of this can be found in the program (DataCollectionProgram.py - lines 1381 through 1383). Finally, dividing the number of unique tokens by the number of words in the text results in the text's lexical richness, and provides an indication of how diverse the word use in a text is. With regard to the goal of the present study, each of those features can be useful for identifying one class of text from another. The word count is slightly less useful in this particular case, as all 500 sample

texts in the Brown Corpus are 2000 words long, but the unique token counts and lexical richness values can indicate whether, for example, words are repeated often. This, as suggested by Koppel et al.'s results (2002), could be an indication of a genre that has that trend.

The second part of the data collection concerns the most frequent words that were found in the texts. The approach used was akin to a Bag-of-words approach, which attempts to distinguish texts, or classes of texts based on the most frequently occurring words in a text. In order to achieve this, a number of most frequent words for each of the 15 classes in the Brown Corpus had to be collected. This was achieved with the NLTK function *FreqDist*, or Frequency Distribution. After it was set up, running the line *'fdist.most_common(20)'* printed the 20 most commonly occurring words and their frequencies (DataCollectionProgram.py – line 1175). Using the program, the 20 most common words for each genre were collected. Naturally, there was some overlap. After accounting for this, the 300 words combined to a total of 107. The full list of 300 words can be found in Appendix A. The list of 107 words with the number of overlaps can be found in Appendix B. Once the most common words had been established, the texts had to be individually checked for them. The command that the program was given boiled down to: check each word in the texts, if the word is one of these words, add a 1 to the variable of the corresponding word, and move on to the next word. The program performed exactly as instructed. Finally, because multiple texts did not contain one or more of the most common words, the variables for each word had to be smoothed. The smoothing applied was Laplace smoothing, or additive smoothing. This entailed adding a value of one to each variable. This smoothing of the data was a necessary form of skewing, as the classifier that would be used could not properly deal with values of zero. The reason for this will be explained later.

Regardless, this process was not required for any feature other than those collected by this Bag-of-words approach.

3.4.2 – Emotional Features

The third part is the collection values for the Pleasantness, Activation, and Imagery features of the Dictionary. This part was considerably more challenging, as it involved a file from the Dictionary that contained a two dimensional data frame that had four columns and just shy of 9000 rows. The first of the four columns contained the words, while the remaining three contained the Pleasantness, Activation, and Imagery values. An example of the Dictionary can be seen in Table 2.

Table 2 – Example of Whissell's Dictionary of Affect in Language (2009)

Word	Pleasantness	Activation	Imagery
a	2	1.3846	1
abandon	1	2.375	2.4
abandoned	1.1429	2.1	3
abandonment	1	2	1.4
abated	1.6667	1.3333	1.2

As can be seen, each row contained a word and its values. The challenge was that the program would have to go through each text and check whether the word was in the first column of the file. If it was not, it could skip the word and check the next word. If it was, it would then have to check what the word's affective values were and add the values to variables that would contain the sums of those values. Moreover, every time a word matched a word in the Dictionary list, it would be tallied. The program would use this to calculate the average Pleasantness, Activation, and Imagery scores for each text. With these goals in mind, the required code that collected the necessary data was written, which can be seen in Figure 1.

```
d = Counter(data[np.isin(data, df.word)])
pleasantness, activation, imagery = (0,0,0)
for k,v in d.items():
      values = df.loc[df.word == k]
      pleasantness += values["pleasantness"].item()*v
      activation += values["activation"].item()*v
      imagery += values["imagery"].item()*v
dict_count = sum(d.values())
p_avg = pleasantness / dict_count
a_avg = activation / dict_count
i_avg = imagery / dict_count
```

Figure 1 - Data collection for Dictionary (DataCollectionProgram.py - lines 1731 through 1743)

The fourth and final part is the collection of values for the features from the EmoLex. There was, however, a singular and rather extensive issue with this lexicon. The list downloaded from the EmoLex's website, which covered all words and all emotions, contained duplicates of words, which would heavily skew the results. On further investigation, it appeared that the duplicates were the result of different combinations of the same emotion. An example of this is shown in Table 3.

Table 3 - Four lines from the EmoLex showing the duplicate entries for the word 'abhor'

Anger	Disgust	Emotion	Fear	Negative	Word
anger	disgust	anger	fear	negative	abhor
anger	disgust	disgust	fear	negative	abhor
anger	disgust	fear	fear	negative	abhor
anger	disgust	negative	fear	negative	abhor

The rightmost column contains the word whereas the other columns contain the emotions (the columns for the remaining emotions were removed as they were empty). The file was structured in such a way that there was a general emotion column (third column), as well as a column for each separate emotion (first, second, fourth, and fifth), which resulted in a unique row for each

combination. As a result, a workaround had to be created. Ultimately each wordlist for the individual emotions was acquired separately. This allowed for an approach similar to the one used for the Bag-of-words features. The program went through each text and tried to find a match for each word in each of the lists. If a word was found, it added a value of one to the respective variable. It then divided the variables by the total number of words that were found in the EmoLex, which produced the ratio between each emotion and the total number of words found. An example for the code that collected the data for the 'anger' emotion can be seen in Figure 2.

```
d_anger = Counter(data[np.isin(data, df_anger.word)])

anger = [sum(d_anger.values()) / len(data)]
```

Figure 2 - Data collection for EmoLex - Anger (DataCollectionProgram.py - lines 1388 and 1780)

The processes described in the two above sections were combined in a final piece of code which took each variable and printed it to a new row of a csv file, which it would output once all texts had been processed (DataCollectionProgram.py – lines 1881 through 1899. This csv file thus became the dataset used for both the classification processes as well as all other forms of testing.

3.5 – Testing

Once the data collect program was completed, it was run. It produced a two dimensional data frame consisting of 501 rows (including header) and 126 columns. As the goal of this study was to determine whether a combination of lexical and emotional features would result in superior classification when compared to either type of data used separately, another program had to be created that would perform as the classifier. Therefore, a choice had to be made with

regard to which classification algorithm would be used. There are a number of algorithms that could be employed in this situation. For instance: Support Vector Machine, Hidden Markov Model, Decision Trees, etc. However, as stated by Wolpert (1996), there is no one classification algorithm that performs best across all tasks. Furthermore, it is impossible to know for certain how well a classifier will perform, prior to testing. As such, the decision in this particular case was influenced largely by convenience and simplicity.

The classification algorithm this study ended up employing was the Naïve Bayes classifier. The theory upon which the Naïve Bayes classifier is based is called Bayes' Theorem. It is named after Thomas Bayes (1764) who first acknowledged and described its utility. The Theorem works on the concept of conditional probability, a concept that determines the probability that an event will occur, given that another event has already happened. The formula for calculating this probability is formulated as follows:

$$P(H \mid E) = \frac{\mathrm{P(E \mid H)} * \mathrm{P(H)}}{\mathrm{P(E)}}$$

Where: P(H) represents the probability of hypothesis H being true; P(E) represents the probability of the evidence; P(E|H) represents the probability of the evidence given that hypothesis is true; and P(H|E) represents the probability of the hypothesis given that the evidence is present. The Naïve Bayes classifier uses this Theorem to calculate the probabilities of a sample belonging to a certain class while assuming that each given feature is independent from one another. In other words, while going through the data frame, the classifier will assume that each feature, which is represented by a column with values for each row, is in no way related or linked at all to any of the other features. Hence the 'Naïve' in its name. In the present study, it was used to calculate the probability that a sample belonged to one of the 15 classes. It tested the

hypothesis of a sample belonging to a certain class by using the values provided in the feature columns as multiple, independent evidences. This changed the formula to the following:

$$P(H\ |En) = \frac{\mathrm{P(E1\ |H)} * \mathrm{P(E2\ |H)} * \mathrm{P(E3\ |H)} * \ \ldots\ \mathrm{P(Ei\ |H)} * \mathrm{P(H)}}{\mathrm{P(En)}}$$

Where: P(En) represents the combined probability of the multiple evidences and the numbers in P(E1, 2, etc.) and 'i' in P(Ei) represent the individual evidences. This, however, can be problematic.

As can be seen in the formula, the probability of each independent feature given that the hypothesis is true is multiplied by the same probability of another independent feature, which can present an issue in text classification. For example, the Bag-of-words approach took the 20 most common words from each text, combined them, and subsequently checked each text for the frequency in which those words occurred. Not all texts will contain at least one of each of those words, resulting in a value of zero. If this remains unchanged and the classifier uses that value for calculating a probability, the result will always be 0%. This is, naturally, incorrect, but it is a persistent issue that needs resolving before the classifier can be considered accurate. A solution to this, which had been applied in the present study as mentioned in section 3.4.1, is to apply smoothing. By adding a value of 1 to each of the features in the bag-of-word approach, the classifier would function without issue. Because the added value is relatively low and applied across all texts, it should not skew the results in any noteworthy manner.

The main reason for choosing this classifier was because it was the most frequently mentioned classifiers when researching the topic of text classification within the Python community. Furthermore, writing the code required to create a program that employs the classifier was particularly simple, as can be seen in Figure 3.

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(x_train, y_train)

y_pred = classifier.predict(x_test)
```

Figure 3 - Code for Naïve Bayes classifier (Classifier.py – lines 58 through 62)

As can also be seen in Figure 3, a specific type of the Naïve Bayes classifying algorithm was used. There are various types of the algorithm, each with a specific type of data in mind. For text classification, a commonly used type is the MultiNomial Naïve Bayes. The reason for this is because text classifications frequently include the Bag-of-words approach, or something similar. Such approaches produce features such as the frequency of certain words. In other words, it produces discrete, whole numbers, which the MultiNomial algorithm is optimised for. However, while the present study indeed produces discrete values for the lexical features, it also produces continuous values for the emotional features. Even though the MultiNomial algorithm performs far superior when it comes to classifying based on discrete values, it is incapable of properly classifying based on continuous values and does not function at all when presented with negative values (Appendix C). As such, the present study employs the Gaussian Naïve Bayes algorithm.

Finally, the Naïve Bayes is a graphical model based classifier, meaning that it does not deal with distances or similarities in the form of a scalar product (such as SVM) and thus does not change when the data is scaled or standardised. As such, and because the present study dealt with various different types of features, the data was scaled prior to being classified. The code used for the scaling can be seen in Figure 4.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

Figure 4 - Code for scaling data (Classifier.py – lines 49 through 52)

What this scaling does is calculate the mean for each individual feature and subsequently gives each value a new value based on its distance from that mean in standard deviations. For instance, if a feature has a mean of 5 and a standard deviation of 1, a value that was initially 5.5 became 0.5 after scaling. The result is that all values across all features are standardised. Again, this should have been neither necessary nor impactful when employing the Naïve Bayes classifier. However, it might have prevented any unforeseen issues that could have been caused by the different data types employed in the study.

Prior to running the texts through this classifier, however, the texts and their values had to be split up into training and testing data. For each run, 70% of the 500 texts were used to train the classifier. The classifier would use the values provided by the training data to establish which combination of values are most indicative for each class. The remaining 30% was used as testing data. Once the classifier had trained itself, it would check the values for each of the texts in the testing data and calculate which class it belonged to. This splitting of the data was done automatically with the code shown in Figure 5.

```
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.30,
random_state = 0)
```

Figure 5 - Code for splitting texts up into testing and training data (Classifier.py – lines 42 and 43)

Also shown in Figure 5 is that the function used for splitting the texts up also had the attribute ‘random_state’. The value given to this attribute determined the manner in which the splitter function split the texts. In other words, entering the value ‘5’ would ensure that every time the program was used, the splitting of texts into training and testing data was exactly the same. In other words, any run done with the attribute set to 5 would be done with the exact same selection of rows. This was particularly useful due to the many different combinations of the sets of features that were tested, which will be discussed further in the final paragraph of this section. In

order to get a proper indication of how well the classifier would perform while employing the Brown Corpus, four tests were run using the automatic text splitter function. Each run had its own value for the 'random_state' attribute. These values were obtained using a random number generator, which would provide a random number between 0 and 10,000. Aside from these four runs, one more run was done, which used manually split data. The reason for this was because some of the classes had a small number of texts. For instance, the Science Fiction class had a total of six samples. As a result of this unbalanced distribution, an automatic split of the texts could potentially result in none of the six samples from the Science Fiction class being used as training data. This in turn would result in there being no chance that any of the six samples would be classified correctly. To avoid this issue and to determine what would happen by doing so, one run was done with a manual split. This manual split was created by simply choosing the first 70% of texts for each class as training data and the remaining 30% as testing data. For instance, if a class had ten samples, the first seven would be used as training data and the last three as testing data. Although this approach had its own potential issues (there was no certainty with regard to whether the first 70% of texts of each class were representative of all texts of their classes), it did ensure that each class was represented properly in both the training and testing data. The order of the texts that was used was the same as how the texts were ordered in the Brown Corpus. How each individual run performed compared to the other runs will be shown in the following section.

As mentioned, various combinations of the multiple sets of features detailed in the previous section were tested. These combinations allowed for determining not only which individual sets of features were most effective, but also which combinations would prove fruitful. Due to the short span of time allotted for the present study, the combinations were

limited to the sets of features. It was unfeasible to analyse the performance of individual emotional features, as it would same be too time-consuming. As such, and as detailed in the previous section, the sets were split up as can be seen in Table 4.

Table 4 - Summary of the contents of each set of features

General:	Word count, token count, and richness
Emo-Dict:	Pleasantness, activation, and imagery
Emo-Emolex:	Anger, anticipation, disgust, fear, joy, negative, positive, sadness, surprise, and trust
Words:	All 107 words from the Bag-of-words approach

All possible combinations between these four sets were made, in an attempt to be as comprehensive as possible. Although they are likely quite obvious, the combinations will nevertheless be listed in Table 5 below. This is done in order to avoid any confusion, as the names of the combinations will be referred to in the next section.

Table 5 - The contents of each combination

Combination name	General	Emo-Dict	Emo-Emolex	Words
Emo-Dict		X		
Emo-Emolex			X	
Emo-Full		X	X	
Words				X
General	X			
Emo-Full-Words		X	X	X
Emo-Dict-Words		X		X
Emo-Emolex-Words			X	X
General-Emo-Full	X	X	X	
General-Emo-Dict	X	X		
General-Emo-Emolex	X		X	
General-Words	X			X
General-Emo-Dict-Words	X	X		X
General-Emo-Emolex-Words	X		X	X
Full	X	X	X	X

This, combined with the five different splits of training and testing data, resulted in 75 runs of the classifier program. However, it would also be useful to determine the effect that sample size might have on the classifier. For that reason, another 75 runs (with the same splits) of the program were done. These runs used two classes: fiction and non-fiction. The former consisted of 125 texts and the latter of 375. All told, there were a total of 150 runs, the results of which will be shown in the following section. For ease of reference, the runs done with all 15 classes will be referred to as the first test and the runs done with just two as the second test.

4 – Results

4.1 – Classification

Each of the 150 runs resulted in a confusion matrix, a visual presentation of the number of true positives, true negatives, false positives, and false negatives that the classifier produced. Alongside the confusion matrix, the program also provided a summary that displayed the precision, recall, and F1 scores of that particular run. An example can be seen in Table 6 and Figure 6.

class	precision	recall	f1-score	support
1	0.27	0.23	0.25	13
2	0.25	0.50	0.33	8
3	0.25	0.20	0.22	5
4	0.12	0.40	0.19	5
5	0.42	0.45	0.43	11
6	0.21	0.27	0.24	11
7	0.38	0.14	0.20	22
8	0.20	0.56	0.29	9
9	0.50	0.04	0.08	24
10	0.20	0.11	0.14	9
11	0.00	0.00	0.00	7
12	0.00	0.00	0.00	2
13	0.32	0.67	0.43	9
14	0.33	0.33	0.33	9
15	0.00	0.00	0.00	3
avg / total	0.30	0.25	0.22	147

Table 6 - Summary of results for run Full for manual split, 15 classes, split: manual

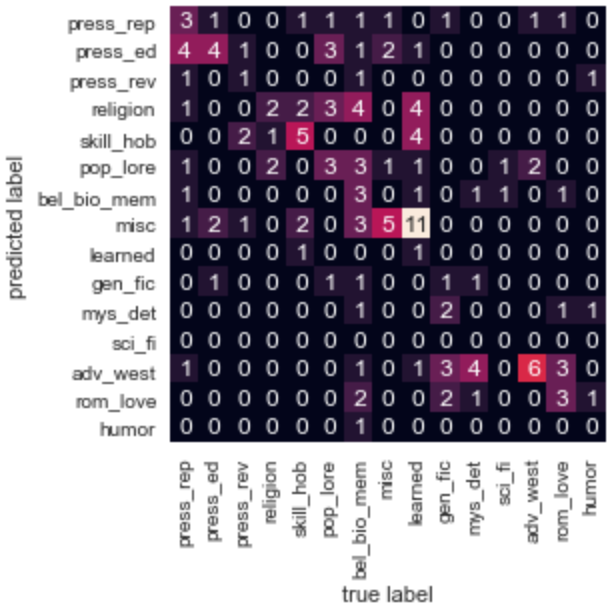


Figure 6 - Confusion matrix for run Full for manual split, 15 classes, split: manual

As mentioned, the results provided in the summaries indicate the precision, recall, and F1 scores of each run. The first column of the summary, as shown in Table 6, indicates each class; there are a total of 15 classes. Comparing the summary to the confusion matrix in Figure 6 shows that class 1 stands for the class ‘press_rep’, or Press Reportage as it is indicated in the Brown Corpus, 2 stands for ‘press_ed’, or Press Editorial, etc. The final column, support, indicates the number of texts that the program classified. In other words, those are the number of texts that were assigned to the testing data after the splitting process. The remaining three columns show the precision, recall, and F1 scores. These are measures of the classifier’s performance. They are calculated using the number of true positives, true negatives, false positives, and false negatives. The reason for this is because those four types of results indicate the number of correct and incorrect decisions the classifier made, as well as the type of decision.

Every confusion matrix has an imaginary diagonal, which, in this case, runs from the top left to the bottom right. This diagonal shows the ‘true’ decisions that the classifier made, be they negative of positive. The true positives are the number of correct positive decisions made by the

classifier. When a text is from the class ‘press_rep’ and it is classified as ‘press_rep’, it is a true positive. The cell containing the true positives is also required to find the remaining three numbers. If you were looking at a specific cell in the diagonal line, the class on the true label axis would be the class of interest. While the number of true positives is a specific cell on the diagonal line, the number of true negatives is the sum of the remaining values on the diagonal line. True positives for one class are seen as true negatives for another. True negatives are the texts that are correctly predicted as being the true positives for a different class, for they do not belong to the class of interest. The false positives and false negatives are most easily explained using the confusion matrix shown above (Figure 6). For each cell containing the true positives of the class of interest, the false positives are the values in the cells in the same row, while the false negatives are the values in the same column. The false positives are the number of texts from other classes that were incorrectly classified as the class of interest. The false negatives are the number of texts from the class of interest that were incorrectly classified as a different class (Lantz, 2015). In other words, if we consider the class ‘press_rep’ to be our class of interest, it has 3 true positives (TP), 34 true negatives (TN), 10 false negatives (FN), and 8 false positives (FP). These numbers can then be used to calculate the measures of performance.

Precision is the ratio of correct positive predictions to the total number of positive predictions. It answers the question: of all texts that were predicted to be a certain class, how many actually belonged to that class? A high precision value would relate to a low number of false positives. Recall is the ratio of correct positive predictions to the total number of texts that actually belong to the class of interest. It answers the question: of all texts that were predicted to be a certain class, how many should have been predicted to be that class? A high recall value

would thus relate to a low number of false negatives. The ratios are therefore calculated as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

The final measure of performance, the F1 score, is the weighted average of precision and recall. While the latter two indicate the performance of specific parts of the classifier, the F1 score gives an indication of how well the classifier performs in more general terms. For instance, class 9 in the table above (Table 6) has a relatively high precision, but quite low recall scores. As a result of the latter, the F1 score is subsequently far closer to the precision score than the recall score. F1 scores can be calculated using the following formula:

$$F1\ Score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

The F1 scores for the four feature sets across all classes can be seen in Table 7. Worthy of note is that the F1 scores can differ quite a bit between the feature sets, to the point that some won't have managed to classify any of the texts of a certain class correctly.

Table 7 - Average F1-scores across all 5 runs with 15 classes

Class	Dictionary	EmoLex	General	Words
Press_rep	0.302	0.298	0.468	0.578
Press_ed	0.314	0.262	0.122	0.292
Press_rev	0.234	0.338	0.314	0.302
Religion	0	0.196	0	0.216
Skill_hob	0.29	0.266	0	0.346
Pop_lore	0	0	0.164	0.25
Bel_bio_mem	0.302	0.284	0.366	0.13
Misc	0	0.31	0.216	0.364
Learned	0.56	0.306	0.458	0.282
Gen_fic	0.216	0.026	0	0.124
Mys_det	0.154	0.232	0.27	0.13
Sci_fi	0	0.196	0	0
Adv_west	0.21	0.302	0	0.378
Rom_love	0.158	0.114	0.136	0.176
Humour	0	0	0.2	0.05

Also interesting is that the F1 scores can differ considerably depending on the feature set used. This suggests that some features are better for certain classes than others. However, such differences are far easily determined when visualised.

As such, the results for each type of run across all five different splits were taken and used to calculate the average for each combination of features. These averages were subsequently visualised in Figure 7.

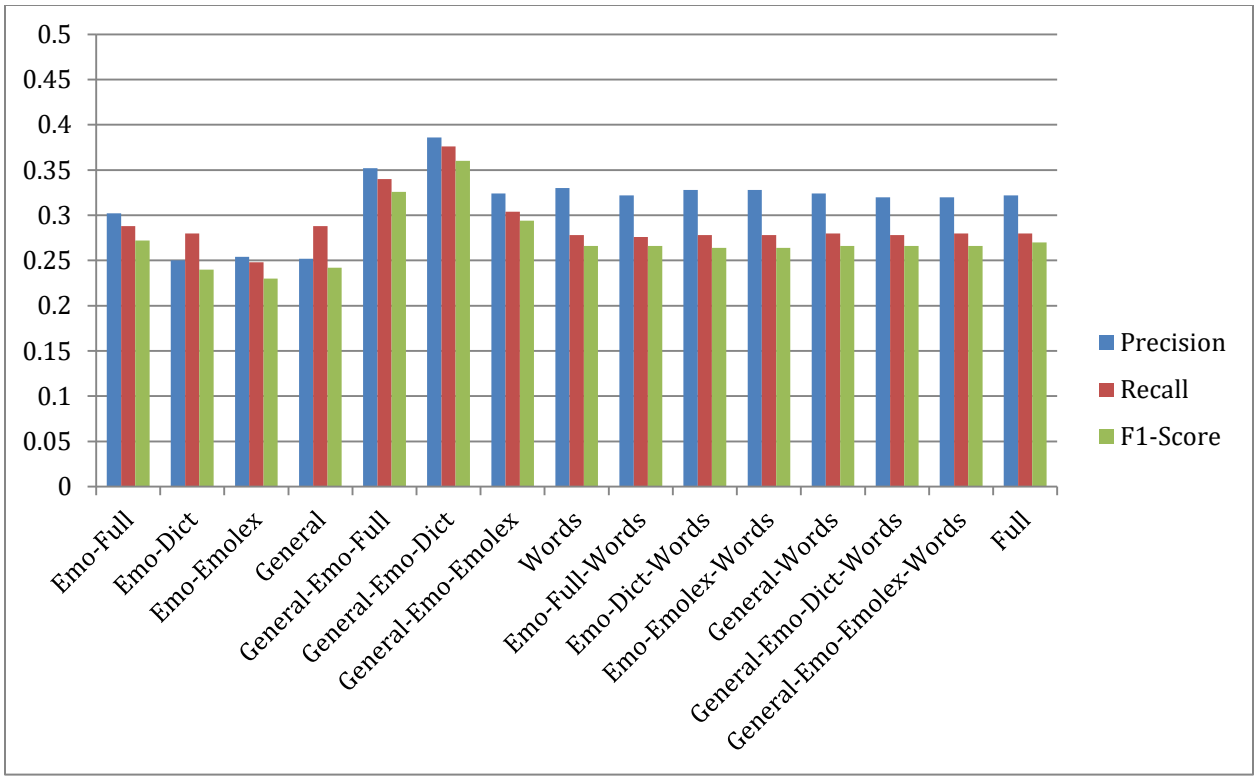


Figure 7 - Results of 15 Gaussian NB runs over 5 different splits with 15 classes

Please note that the maximum value for the y-axis is 0.5. For more exact values, please refer to Table 8 below.

Table 8 - Results of 15 Gaussian NB runs over 5 different splits with 15 classes

Feature Combination	Precision	Recall	F1-Score
Emo-Full	0.302	0.288	0.272
Emo-Dict	0.25	0.28	0.24
Emo-Emolex	0.254	0.248	0.23
General	0.252	0.288	0.242
General-Emo-Full	0.352	0.34	0.326
General-Emo-Dict	0.386	0.376	0.36
General-Emo-Emolex	0.324	0.304	0.294
Words	0.33	0.278	0.266
Emo-Full-Words	0.322	0.276	0.266
Emo-Dict-Words	0.328	0.278	0.264
Emo-Emolex-Words	0.328	0.278	0.264
General-Words	0.324	0.28	0.266
General-Emo-Dict-Words	0.32	0.278	0.266
General-Emo-Emolex-Words	0.32	0.28	0.266
Full	0.322	0.28	0.27

As can be seen in both Figure 7 and Table 8, the classifier performed better when using both the emotional feature sets combined than when they were used individually. Also noteworthy is that any feature combination that included the Bag-of-words features resulted in roughly the same performance. This includes the ‘Full’ run, which employed all four sets of features. Finally, the combination that performed best was the one that used both the General and the Dictionary features.

As mentioned in the previous section, a second round of classification was done using only two classes (fiction and non-fiction) to determine the effect that sample size might have on the performance of the classifier. The results of this test have been visualised in Figure 8.

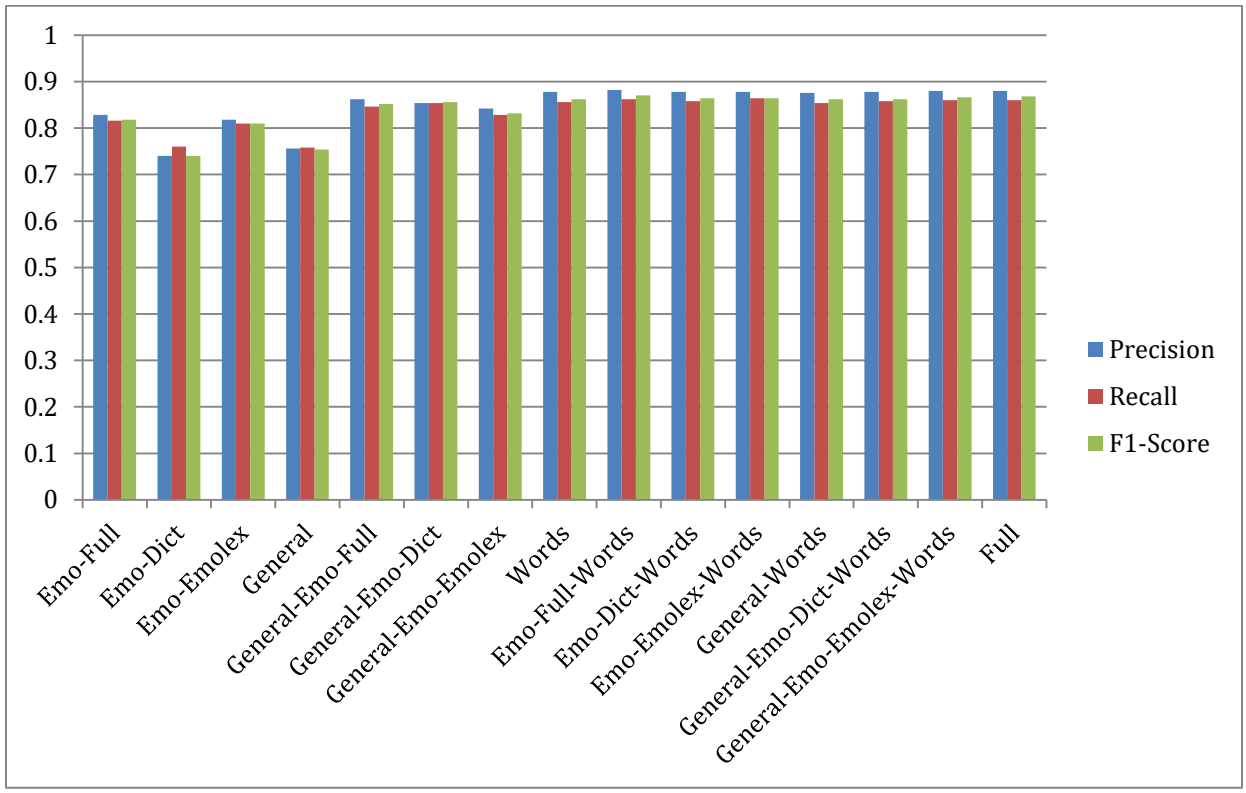


Figure 8 - Results of 15 Gaussian NB runs over 5 different splits with 2 classes; non-fiction and fiction

Please note that, as opposed to the previous Figure, the maximum value on the y-axis for this graph is 1. For more exact values, please refer to Table 9 below.

Table 9 - Results of 15 Gaussian NB runs over 5 different splits with 2 classes; non-fiction and fiction

	Precision	Recall	F1-Score
Emo-Full	0.828	0.816	0.818
Emo-Dict	0.74	0.76	0.74
Emo-Emolex	0.818	0.81	0.81
General	0.756	0.758	0.754
General-Emo-Full	0.862	0.846	0.852
General-Emo-Dict	0.854	0.854	0.856
General-Emo-Emolex	0.842	0.828	0.832
Words	0.878	0.856	0.862
Emo-Full-Words	0.882	0.862	0.87
Emo-Dict-Words	0.878	0.858	0.864
Emo-Emolex-Words	0.878	0.864	0.864
General-Words	0.876	0.854	0.862
General-Emo-Dict-Words	0.878	0.858	0.862
General-Emo-Emolex-Words	0.88	0.86	0.866
Full	0.88	0.86	0.868

Immediately apparent is the fact that the performance scores are far higher when classifying the samples into just 2 classes. As was the case in the first test, when using both the sets features contained in the Dictionary and the EmoLex to classify the samples, the combination performed better than when using sets of features separately. Moreover, as was also the case in the first test, all runs that included the Bag-of-words feature set result in nearly identical F1 scores. When the Bag-of-words feature set was used together with the emotional feature sets, the classifier performed only marginally better. Interestingly, while the combination of the General and Dictionary feature sets resulted in the highest performance scores in the first test by a noticeable degree, they are slightly lower than the runs that included the Bag-of-words features. Finally, the relatively low performance scores for the runs of the emotional features in

the first test were understandable due to the low sample size for each class. However, the fact that they were also among the lowest in the second test was enough to warrant further investigation.

In light of this, an initial analysis of the values of the emotional features was done. For the features from the Dictionary, an approach was used that the Dictionary's creator also used. Whissell (2009) employed recordings of speech to demonstrate how the Dictionary could be used to analyse differences in language. After extracting the Pleasantness and Activation scores from the recordings, she turned the scores into vectors. These vectors she subsequently visualised, which can be seen in Figure 9. The speech analysed in Whissell's study was recorded during an incident on the 19th of April, 2000, which involved the police and a man who had barricaded himself in a house. During the incident, which lasted for several hours, he spoke with the police on multiple occasions. Whissell analysed his speech using the Dictionary and found that over the course of those hours, the language used by the man changed noticeably.

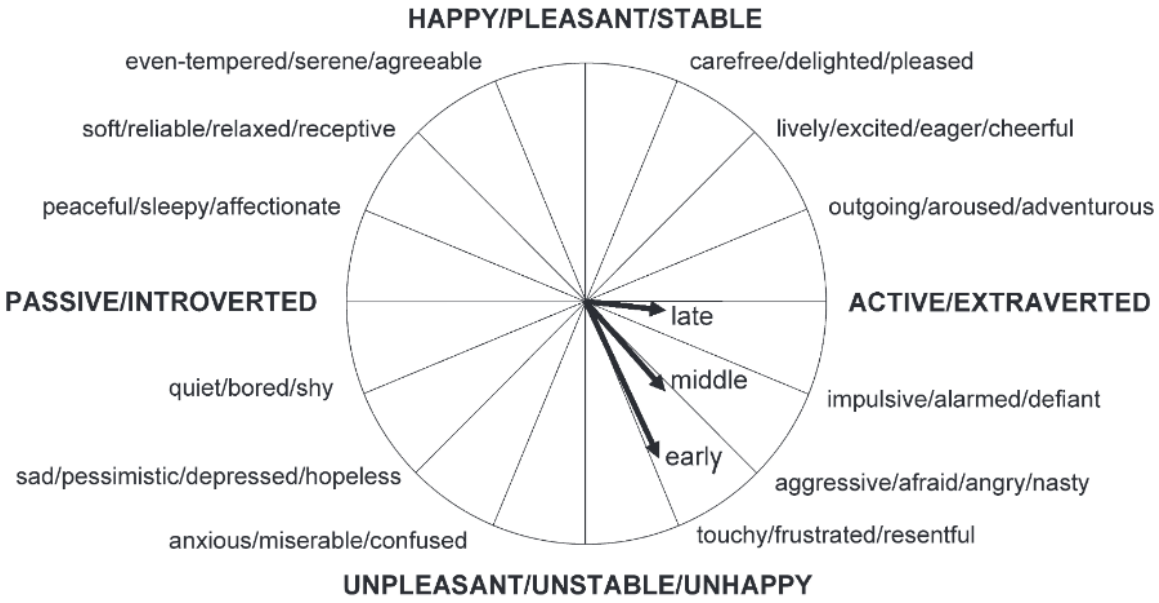


Figure 9 - "Emotional vectors describing the undertones of language used by a shooter in the early, middle, and late stages of a standoff" (Whissell, 2009, p. 519)

This demonstrates that the Pleasantness and Activation features can demonstrate differences between types of language use. It should, however, of course be noted that Whissell's application of this approach indeed used spoken language that was uttered in a highly stressful time, thus likely evoking more emotional use of language. However, it stands to reason that it ought to be indicative of the features' ability to differentiate between certain types of written language use, as well.

First, in order to compare with Whissell's own results, the results of the present study were used to determine the matching rate. In the present study, the Dictionary had an average matching rate of 76.3%. Subsequently, the data obtained through the first program for the Pleasantness and Activation features were used to create the visualisation in Figure 10.

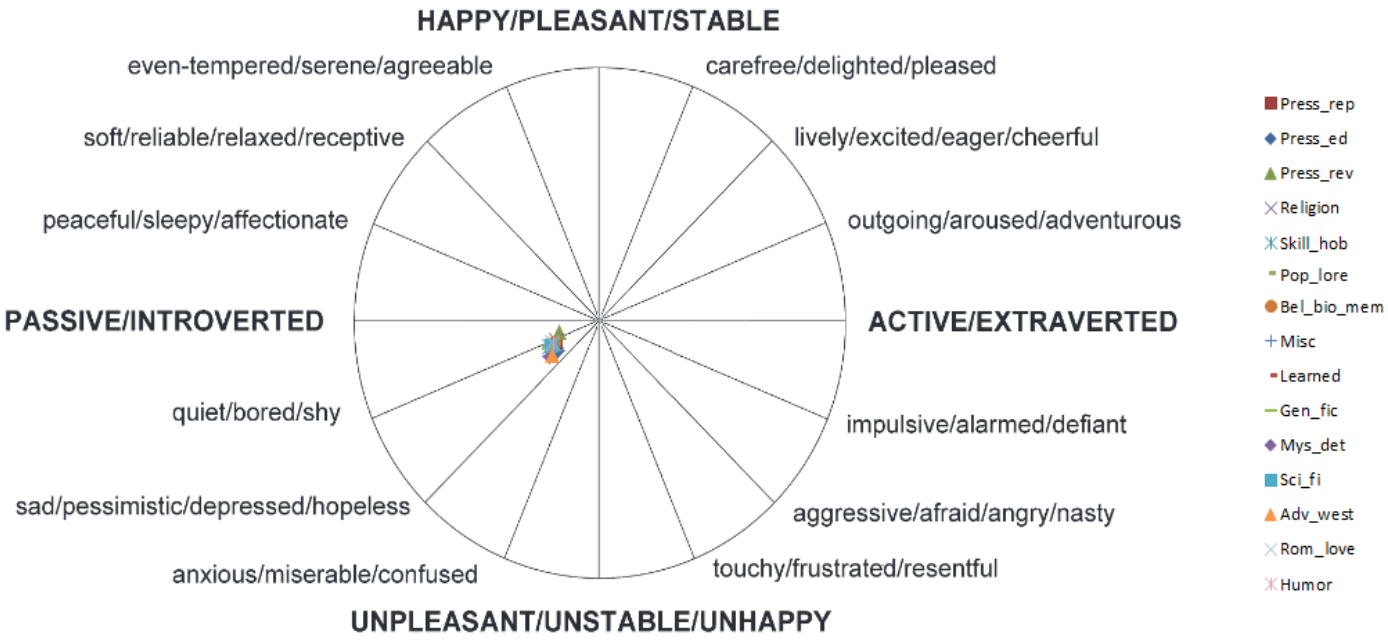


Figure 10 - Scatterplot of Pleasantness and Activation per class

Immediately apparent is that the vectors for each of the 15 classes converge quite close together. In fact, they are so close to each other that it is challenging to discern the position of each class without looking at the actual values (Results.xlsx - Sheet: 'Emotion avg. per class'). As such, further, statistical examination is required to determine the validity of using the Dictionary features. For instance, statistical analyses are required to determine whether the results between the classes differ significantly from one another. If they do, they can be used to distinguish classes from one another. If they do not, it is likely that the features are not suitable for classification purposes. However, prior to conducting those examinations, the results for the EmoLex features ought to be examined first.

As mentioned, the results for the EmoLex features came in the form of ratios. Each value is a ratio of how many words containing that emotion were present in the text to the total number of words. As can be seen in Table 10, the total ratio values differ when compared between classes. The reason for this is because not every class had the same number of matches in the

EmoLex. For the 'Press_rep' (Press: Reportage), class, for instance, 39.9% of the words contained in those texts found a match in the EmoLex. Based on these results, the average matching rate of the EmoLex for the present study was 43.2%.

Table 10 - Ratio of emotions according to the EmoLex per class

	Anger	Anticipation	Distrust	Fear	Joy	Negative	Positive	Sadness	Surprise	Trust	Total
Press_rep	0.022	0.046	0.010	0.030	0.031	0.049	0.100	0.023	0.016	0.072	0.399
Press_ed	0.028	0.051	0.015	0.040	0.035	0.070	0.116	0.031	0.021	0.081	0.488
Press_rev	0.022	0.045	0.015	0.029	0.048	0.050	0.118	0.034	0.025	0.062	0.448
Religion	0.033	0.064	0.024	0.052	0.052	0.067	0.141	0.034	0.021	0.093	0.582
Skill_hob	0.017	0.044	0.009	0.022	0.031	0.039	0.105	0.017	0.018	0.058	0.358
Pop_lore	0.027	0.046	0.017	0.036	0.033	0.061	0.106	0.028	0.019	0.068	0.442
Bel_bio_mem	0.028	0.047	0.018	0.038	0.038	0.065	0.116	0.031	0.021	0.070	0.472
Misc	0.013	0.047	0.008	0.027	0.027	0.044	0.129	0.018	0.011	0.091	0.415
Learned	0.020	0.037	0.013	0.030	0.023	0.051	0.098	0.023	0.014	0.059	0.368
Gen_fic	0.028	0.048	0.024	0.036	0.037	0.069	0.084	0.034	0.023	0.051	0.435
Mys_det	0.028	0.040	0.020	0.038	0.027	0.063	0.070	0.032	0.023	0.044	0.385
Sci_fi	0.021	0.053	0.012	0.034	0.036	0.058	0.094	0.029	0.021	0.057	0.417
Adv_west	0.033	0.039	0.024	0.042	0.027	0.075	0.068	0.033	0.023	0.042	0.406
Rom_love	0.026	0.048	0.023	0.034	0.043	0.069	0.085	0.036	0.023	0.049	0.435
Humour	0.026	0.045	0.018	0.035	0.036	0.066	0.094	0.034	0.021	0.056	0.429

This unequal distribution of matches in the EmoLex complicates creating a visualisation showing the differences between the classes. When considered as percentages, however, they provide a decent indication of the distribution of the emotions within the texts. Therefore, the ratio values were converted to percentages based on the total number of words found in the EmoLex for each class. These percentages are shown in Table 11.

Table 11 - Values for emotions in percentages according to the EmoLex per class

	Anger	Anticipation	Distrust	Fear	Joy	Negative	Positive	Sadness	Surprise	Trust
Press_rep	6	12	3	8	8	12	25	6	4	18
Press_ed	6	11	3	8	7	14	24	6	5	17
Press_rev	5	10	3	7	11	11	26	8	6	14
Religion	6	11	4	9	9	12	24	6	4	16
Skill_hob	5	12	2	6	9	11	29	5	5	16
Pop_lore	6	10	4	8	8	14	24	6	4	15
Bel_bio_mem	6	10	4	8	8	14	24	7	5	15
Misc	3	11	2	7	7	11	31	4	3	22
Learned	5	10	3	8	6	14	27	6	4	16
Gen_fic	6	11	5	8	9	16	19	8	5	12
Mys_det	7	10	5	10	7	16	18	8	6	12
Sci_fi	5	13	3	8	9	14	23	7	5	14
Adv_west	8	9	6	10	7	19	17	8	6	10
Rom_love	6	11	5	8	10	16	20	8	5	11
Humour	6	10	4	8	8	15	22	8	5	13

For the purpose of visualising these results for ease of comparison, a stacked area chart (Figure 11) was created based on these percentages.

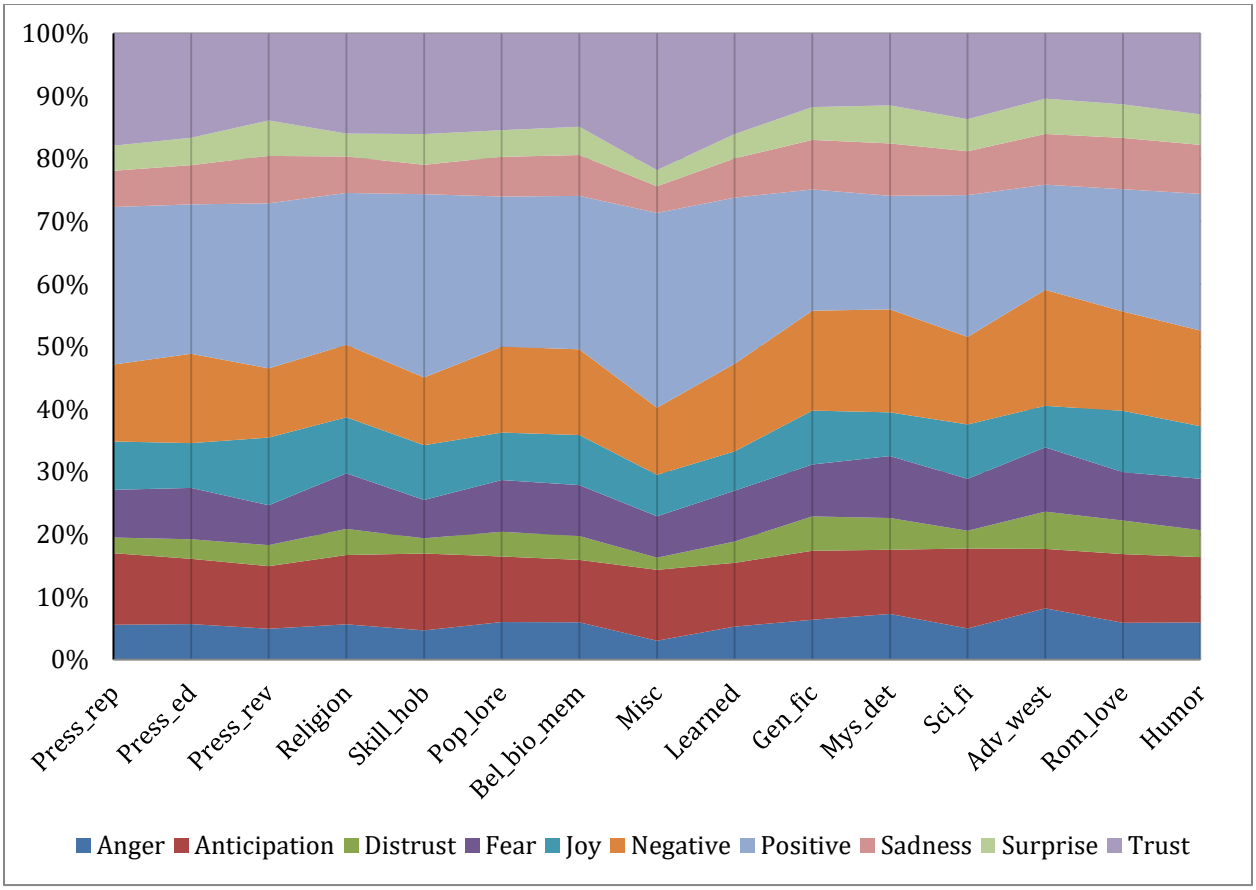


Figure 11 - 100% Stacked Area chart of emotion ratios according to the EmoLex per class

Unlike the visualisation for the Dictionary features, the stacked area chart shows a bit more differentiation between classes. Most notable are the classes ‘Misc’ and ‘Adv_west’, in that they are the highest or lowest in a number of areas. For instance, ‘Misc’ has a high ratio of words related to the emotions Trust and Positive, while it has a relatively low number of words related to the emotions Anger, Distrust, Sadness, and Surprise. ‘Adv_west’, on the other hand, has a relatively high number of words related to the emotions Anger and Negative. However, similarly noticeable differences are not present in all classes. In fact, various classes appear to have a distribution of the emotions similar enough to one another to again warrant a statistical analysis of the features to determine whether or not they are suitable to be used in classification.

4.2 – Statistics

To that end, a few statistical tests were run on the values of the features of both the Dictionary and the EmoLex feature sets. More specifically, both an ANOVA and a post-hoc test would be run on each feature. The ANOVA tests would provide insight into whether a difference in class would result in significant variation between the different emotion features or not. However, ANOVA results are fairly general; unspecific. The post-hoc tests would compensate for that by allowing for a more detailed analysis of the relations between the classes. While an ANOVA would produce just one score of significance across all classes, the post-hoc test would produce a score for each combination of the values of two classes (i.e. Humour - Science Fiction, Humour - General Fiction, Humour - Learned). Aside from these two tests, another test would be done to gain an estimate of the effect size per feature. This estimate would be useful to determine how much of the variance within the features could be attributed to the difference in classes and how much to chance.

The statistical tests were run using SPSS (Version 25; IBM Corp, 2017). Specifically, multiple sets of tests were run. Each contained a general linear model univariate analysis, which used the genres as the fixed factor and a single emotion feature as the dependent variable. This functioned as the ANOVA test. Alongside this analysis, an estimate of the effect size was made in the form of a Partial Eta Squared. Finally, a Tukey's post-hoc test was run. As there were 13 emotional features, a total of 13 sets of tests were run. Furthermore, because the distribution of samples was unbalanced, another 13 sets of tests were run using a random subset of 6 samples per each class. The reason for a subset of 6 samples was because Science Fiction was the class with the lowest number of samples, namely 6.

4.2.1 – ANOVA Results

The results of the statistical tests are awkward to report within the thesis in detail. The reason for this is because they are quite lengthy and thus unwieldy. Because of this, the results displayed within the present study will be limited to summaries. However, the full reports can be found on the GitHub page, mentioned in the Introduction section. Specifically, they are located under root/Results/Statistics.

Table 12 - Results of ANOVA tests and Estimated Effect Size – all samples

Emotional Feature	**ANOVA**	**Estimated Effect Size**
Pleasantness	$F(14,485) = 8.932$, $P < .001$	.205
Activation	$F(14,485) = 4.797$, $P < .001$	.122
Imagery	$F(14,485) = 20.421$, $P < .001$	.371
Anger	$F(14,485) = 6.133$, $P < .001$	.150
Anticipation	$F(14,485) = 4.808$, $P < .001$	.122
Disgust	$F(14,485) = 8.814$, $P < .001$	.203
Fear	$F(14,485) = 4.499$, $P < .001$	.115
Joy	$F(14,485) = 9.007$, $P < .001$	.206
Negative	$F(14,485) = 8.205$, $P < .001$	.192
Positive	$F(14,485) = 15.875$, $P < .001$	.314
Sadness	$F(14,485) = 8.204$, $P < .001$	.191
Surprise	$F(14,485) = 9.794$, $P < .001$	.220
Trust	$F(14,485) = 14.627$, $P < .001$	.297

The results of the ANOVA tests and the estimated effect sizes are shown in Table 12. As can be seen, the p-values for all classes are so low that they indicate significant variance between the values of the features. Furthermore, the F-values are all higher than 1, indicating that there is indeed variance across the group means. Interesting to note, as well, is that the effect sizes indicate that the effect that the classes had varied from medium to large (small = .02, medium =

.13, and large = .26 (Murphy, Myors, & Wolach, 2014)), with an average of .208. The results are comparable with those from the tests run using the subset. Table 13 shows that, even with very few samples, there was still significant variation within the features as a result of the various classes. Although some of the p-values are slightly higher than those in Table 12, they are still well under the alpha level.

Table 13 - Results of ANOVA tests and Estimated Effect Size – subset of 6 samples per class

Emotional Feature	**ANOVA**	**Estimated Effect Size**
Pleasantness	$F(14,75) = 4.128$, $P < .001$	.435
Activation	$F(14,75) = 2.430$. $P < .001$	.321
Imagery	$F(14,75) = 4.478$, $P < .001$	.455
Anger	$F(14,75) = 2.635$, $P = .004$	.330
Anticipation	$F(14,75) = 3.548$, $P < .001$	.398
Disgust	$F(14,75) = 2.260$, $P = .01$	.297
Fear	$F(14,75) = 2.509$, $P = .006$	.319
Joy	$F(14,75) = 4.954$, $P < .001$	.480
Negative	$F(14,75) = 2.948$, $P = .001$	.355
Positive	$F(14,75) = 5.308$, $P < .001$	.498
Sadness	$F(14,75) = 3.619$, $P < .001$	.403
Surprise	$F(14,75) = 2.686$, $P = .003$	.335
Trust	$F(14,75) = 3.664$, $P < .001$	.406

Also interesting to note is that the estimated effect size of the classes is higher when the number of samples is lower. As these results only provide an indication on a general level, the results of the post-hoc tests ought to be analysed as well.

4.2.2 – Post-hoc Results

Table 14 contains a summary of the Tukey post-hoc tests. For a more detailed summary, please see Appendix D, or the full reports in the GitHub respository. Interestingly, certain features have a higher number of significantly different combinations between classes than others. According to these results, when using all samples, the Imagery, Positive, and Trust features result in the greatest number of such combinations, while Activation, Fear, and Anticipation result in the lowest.

Table 14 - Post-hoc results; number of significantly different class combinations per emotional feature

Emotional Feature	No. of significantly different class combinations (out of 210)	No. of significantly different class combinations (out of 210) – Subset of 6 per class
Pleasantness	48	24
Activation	20	8
Imagery	76	24
Anger	32	8
Anticipation	26	12
Disgust	51	4
Fear	20	6
Joy	52	28
Negative	50	8
Positive	84	26
Sadness	53	10
Surprise	52	8
Trust	86	12

Although the values of those features hint at the presence of a trend, there are other values that suggest the opposite. The values for the Negative and Surprise features, for instance. Regardless, the results warranted a test for correlation. As such, a number of tests were run. Using a Shapiro-

Wilk test, both columns were tested for normality. The left column turned out to be normally distributed, while the right one did not. Because of this, both a parametric (Pearson) and non-parametric (Spearman) correlation test were run to determine the relation between the two columns. These resulted in a moderately positive (R = .508 and R = .513, respectively), yet non-significant ($P = .08$) correlation. In other words, there is a large correlation between the differences in the numbers of significant combinations, but it is not significant enough to not to attribute the correlation to chance. However, another trend appears to be present in the above tables.

When comparing Tables 12 and 14, there appears to be a trend between the number of significantly different class combinations per feature and their effect sizes. The features that most noticeably support this trend are: Imagery, Positive, and Trust. These features all have a higher than average effect size and higher number of significant combinations. As such, another correlation test was done. Both columns were normally distributed. As such, a Pearson test was run. This test showed the presence of a very strong positive (R = .942), and highly significant ($P < .001$) correlation. The same was true for the subset. As such, the higher the effect size, the higher the number of significantly different class combinations, and likely the more useful a feature is in distinguishing samples from one class from those of another.

Another interesting set of results can be seen in Table 15. Shown there are the number of samples per class and the number of combinations that showed a significant difference in the post-hoc tests.

Table 15 - Post-hoc results; number of texts vs. significantly different class combinations using all emotional features

Class	Number of Samples	Number of Significant Combinations
Non-Fiction		
Press: Reportage	44	41
Press: Editorial	27	31
Press: Review	17	34
Religion	17	72
Skills and Hobbies	36	56
Popular Lore	48	33
Belles Lettres Biography, Memoirs, etc.	75	44
Miscellaneous	30	72
Learned	80	67
Fiction		
General Fiction	29	37
Mystery and Detective Fiction	24	40
Science Fiction	6	4
Adventure and Western Fiction	29	52
Romance and Love Story	29	40
Humour	9	6

At first glance, there are a few rows that hint at a relation between the two columns. The values for classes such as Press_rep, Learned, Sci_fi, and Humour suggest the presence of another trend. Specifically, the number of samples seems to be related to the number times their values for the emotional features were significantly different from that of other classes. However, the values for Religion and Miscellaneous suggest differently. Therefore, another set of tests was required to check for a correlation between the two. As before, a Shapiro-Wilk test was used to test both columns for normality. This time, the right column turned out to be normally distributed, while the left one did not. As such, both a parametric (Pearson) and non-parametric

(Spearman) test were run again. These tests resulted in a similarly moderately positive (R = .443 and R = .491, respectively), but non-significant (*P* = .26) correlation. There is a medium correlation between the differences in the numbers of significant combinations, but it is again not significant enough to not attribute it to chance.

Finally, the confusion matrices produced by the classifier were compared with the results from the post-hoc test. The possibility existed that there might be a relation between how frequently a certain class combination showed a significant difference and the number of false positives and false negatives shown in the confusion matrices. The hypothesis was that a higher number of significant differences in the post-hoc would relate to a lower number of false negatives and positives produced by the classifier. To determine whether this was the case, the confusion matrices produced by the five runs of the classifier using the 'Emo-Dict', 'Emo-Emolex', and 'Emo-Full' feature sets were added together. Figures 12, 13, and 14 were the final products.

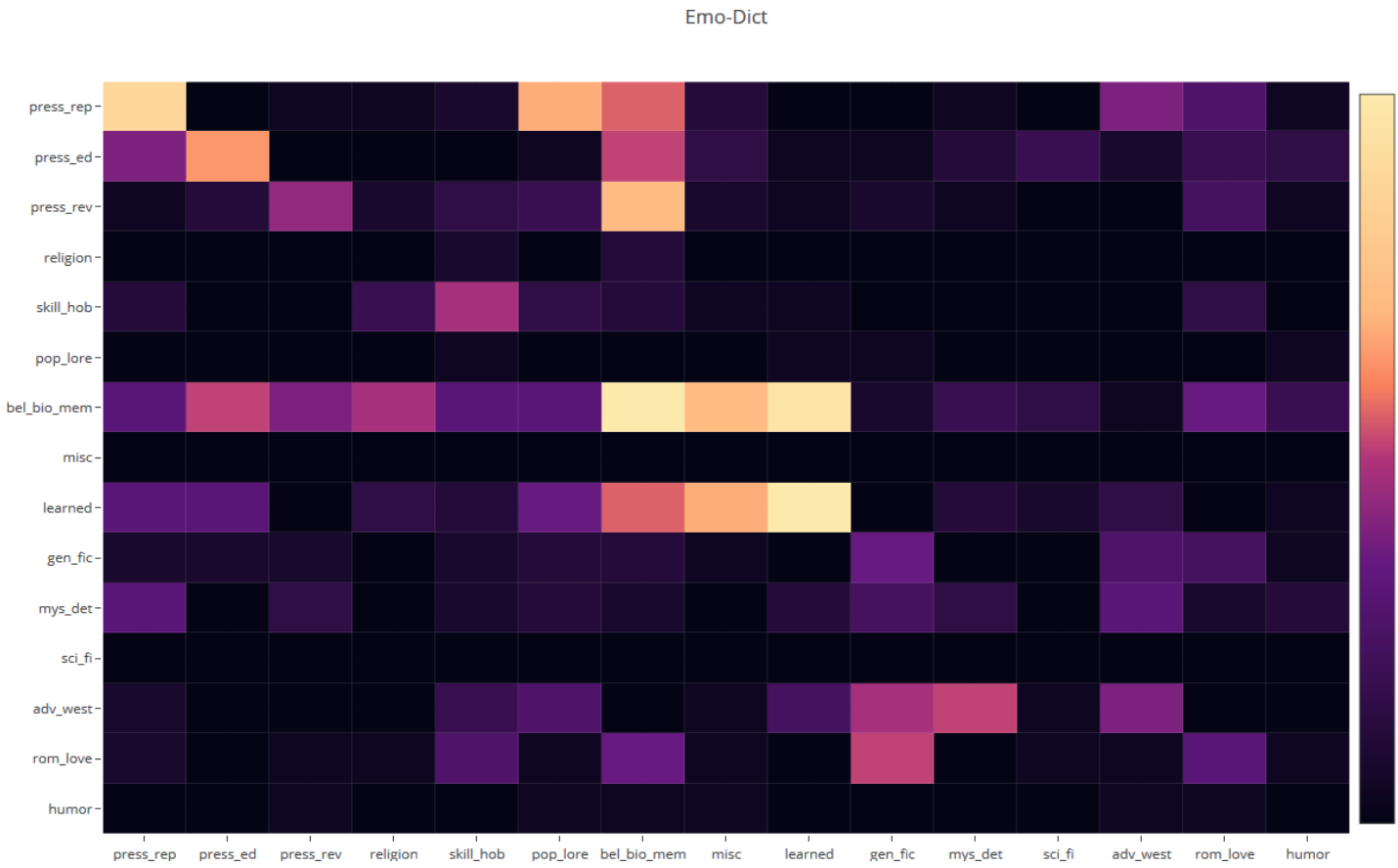


Figure 12 - Combined confusion matrices of five different runs using the Emo-Dict feature set

For the 'Emo-Dict' feature set, for instance, there are four class combinations with a high number of false negatives as can be seen in Figure 12. The values for these cells will be compared with the post-hoc results to see if there is any reason to test for correlation. These four class combination are: 'learned' - 'bel_bio_mem', 'misc' - 'bel_bio_mem', 'misc' - 'learned, and 'bel_bio_mem' - 'press_ed'. The more detailed summary of the post-hoc results shows that these combinations had a total of 1, 0, 2, and 0 significant differences (out of a maximum of 3) respectively (Appendix D).

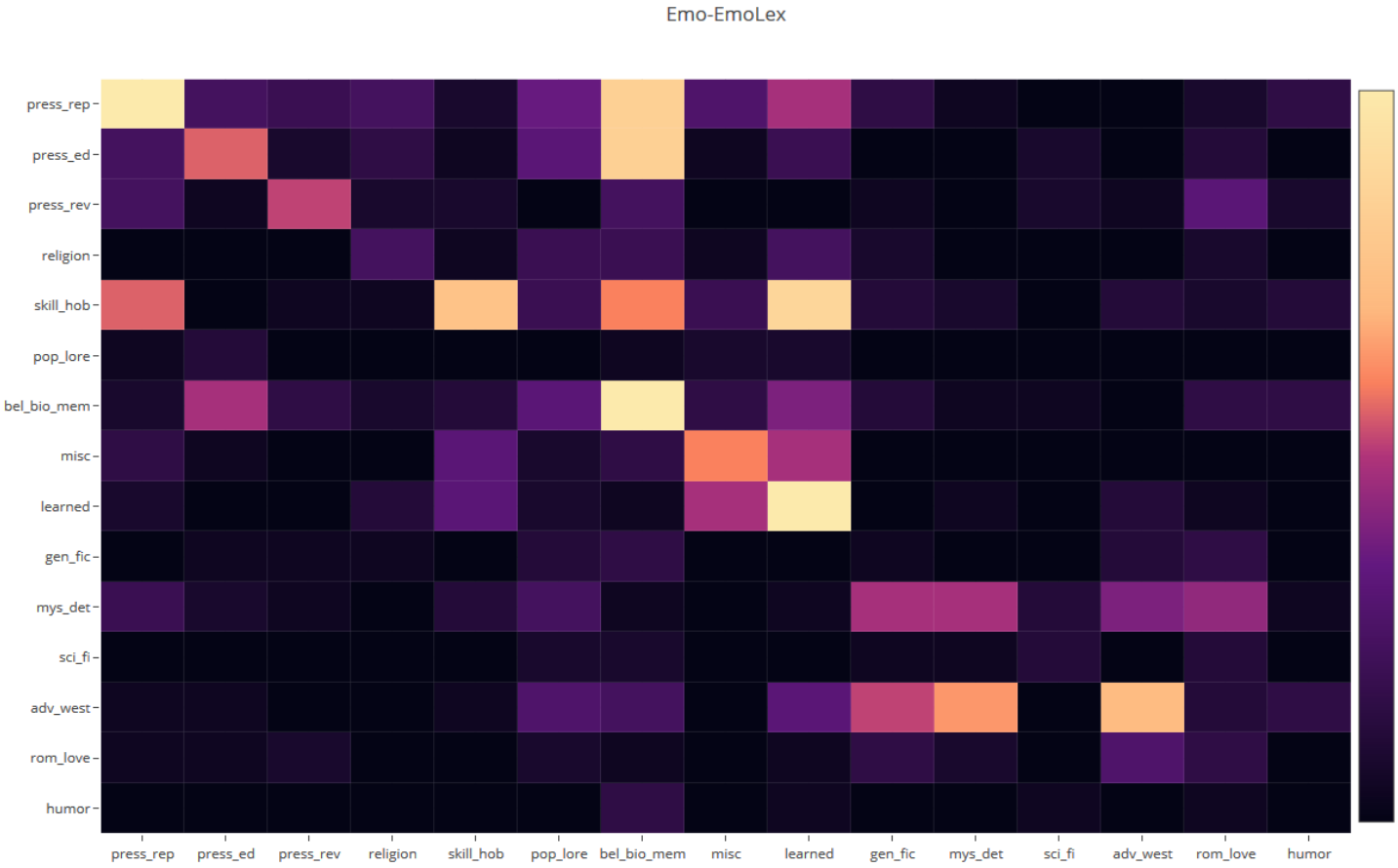


Figure 13 - Combined confusion matrices of five different runs using the Emo-Emolex feature set

The ‘Emo-Emolex’ feature set resulted in the greatest number of false positives and negatives in the class combinations of: ‘press_rep’ - ‘bel_bio_mem’, ‘press_ed’ - ‘bel_bio_mem’, and ‘skill_hob’ - ‘learned’. The results from the post-hoc show that there were a total of 3, 0, and 0 significant differences (out of a maximum of 10) respectively.

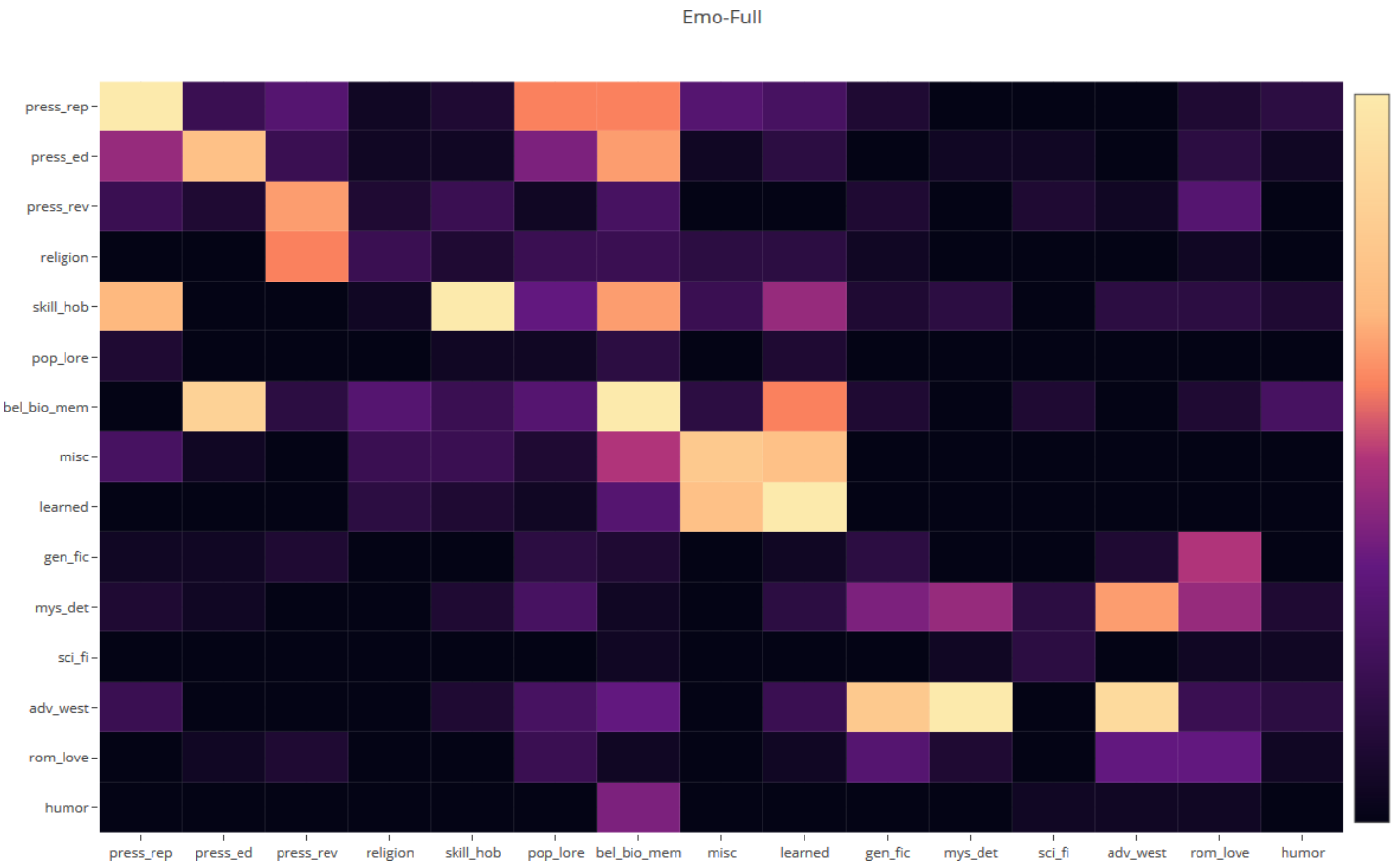


Figure 14 - Combined confusion matrices of five different runs using the Emo-Full feature set

The 'Emo-Full' feature set has a few more class combinations that resulted in a high number of false positives and negatives. The class combinations that will be compared to the post-hoc results are: 'skill_hob' – 'press_ed', 'bel_bio_mem' – 'press_ed', 'learned' – misc', 'adv_west' – 'gen_fic', and 'adv_west' – 'mys_det'. In the post-hoc results, these had a total of 6, 0, 4, 0, and 0 significant combinations (out of a maximum of 13), respectively.

The confusion matrices for all three feature sets contain a few instances of a high number of false positives and false negatives for certain class combinations. However, when comparing these to the number of times that the values of a feature significantly differed between these classes, no trend is found. As such, there does not appear to be an obvious trend. However, in

order to be completely certain that no correlation existed between the two variables, another set of tests were done.

The data used for testing was a combination between the sum total of all five runs per feature set, which resulted in a confusion matrix per feature set that contained all data, and the post-hoc results seen in Appendix D. Please refer to the main results file for the exact data used (Results.xlsx – Sheet: 'PHvsCM'). Shapiro-Wilk tests indicated that all six variables (two variables (confusion matrix & post-hoc results) per feature set) were normally distributed. As such, another three Pearson tests were run. These tests indicated that, for the Dictionary feature set, there was a very weak negative (R = -.087), non-significant (*P* = .21) correlation. For the EmoLex feature set, there was another very weak negative (R = -.025), non-significant (*P* = .72) correlation. Finally, the tests for the Full emotional feature set resulted in yet another weak (R = -.102), non-significant (*P* = .14) correlation. In other words, there did not seem to be a link between the number of times a text was classified as a false positive or a false negative and the number of emotional features that significantly differed between the classes of interest.

5 – Discussion

Overall, the results were not particularly straightforward. The results for the runs with 15 classes suggested that a combination of both emotional and lexical features (General & Dictionary feature sets) is superior to either type being used individually. However, the fact that neither of the feature sets appeared to have any influence when combined with the Bag-of-words features indicated that it is not quite that simple. While the results may have shown the feature sets perform at a certain level, the results did not explain why they did so. Prior to going into those explanations, however, several limitations ought to be addressed.

The first limitation concerns genre. As mentioned, it is unknown what definition of genre was used while compiling the Brown Corpus. However, because the texts were divided up into genres there likely is some relation between the texts of a given genre. Unfortunately, it is unclear what this relation is. Because the definition of genre used while compiling the corpus is known, it is entirely possible that the texts are not sufficiently similar enough with regard to, for instance, the language used in them. This would negatively influence the classifier's performance. The reverse is also possible, which would influence the performance positively. Unfortunately, it is, insofar as can be determined, impossible to find its influence in the results. However, if any of the results would show genre's influence, it would be those of the 'Miscellaneous' or 'misc' genre. As was said previously, 'Miscellaneous' is inherently a genre that contains texts that are not necessarily related to one another in the way texts from other genres are. However, with regard to emotional features, the classifier performed equally well with the 'misc' genre as it did with genres that had a similar number of samples (Table 15 and Figures 12, 13, & 14). In other words, the variance of the results for the samples of the 'misc' genre did not appear to be any greater than those of other genres. Interestingly, its number of

significantly different combinations with other genres was incredibly high for its number of samples (Table 15). The genres ‘misc’ and ‘religion’ were outliers in that regard, suggesting that the affective language used in them was considerably different from that in other genres. Although only ‘misc’ is used as an example here, it is likely that the definition of genre used during compilation had some sort of effect on the results of other genres as well. Ultimately and unfortunately, however, it is not possible to determine what the exact effect was.

The second limitation lies with the emotional lexicons used in the present study. Contrary to what was initially assumed, the EmoLex had a fairly low matching rate when compared to the Dictionary. Indeed, on average 43.2% of the words contained in the Brown Corpus were present in the EmoLex. The high matching rate of the Dictionary as tested by Whissell (2009) was likely because it included words from word classes such as prepositions, articles, and pronouns. However, in the present study, the Dictionary had a matching rate of 76.3%. Considerably lower than the 90% in Whissell’s own study (2009). The reason for this is likely because the stop words were removed from the texts during the pre-processing of the texts. Word classes such as prepositions constitute a considerable part of language. Removing them meant that both the total number of words and the number of words that found a match in the Dictionary decreased. This in turn meant that the percentage of words that did not find a match increased, as they became a more substantial part of the overall number of words. However, while this explains the lower matching rate for the Dictionary, it does not do so for the EmoLex. The EmoLex does not include these word classes. As a result, removing the stop words would actually only have increased its matching rate. This suggests that the matching rate of the EmoLex on regular language requires further investigation.

The way in which the present study was set up did not require the aforementioned word classes. In fact, removing them would make the values for all features more precise as they would not be influenced by words that occurred frequently through all texts. This was the main reason why so few of the features used in Koppel et al.'s study (2002) were employed in the present study. Their study had shown that the frequency with which words from such word classes are used in a text could definitely be indicative of a certain genre. However, using them would have negatively affected the values of the other features. The results suggest, however, that it would likely have been a better alternative to the Bag-of-words approach.

This is seen when looking at the rather homogenous results of all the runs that include the Bag-of-words features. The similarity in those results indicates that the number of features in the Bag-of-words feature set overpowered any other features included in the run. There are a total of 107 features in the bag-of-word feature set and a combined total of 16 in the other three feature sets. This caused the number of features to be thoroughly unbalanced, as it skewed the classification results considerably. The classifying algorithm used calculated the probability of a sample belonging to a certain class based on the values of each individual feature. Because the Bag-of-words feature set contained 87% of all features, the values in that feature set simply overpowered the other features. Because of this, the runs including the Bag-of-words feature set are essentially useless; they only show the performance of that feature set. Unfortunately, that was not the only issue with the setup of the experiment.

It is generally true that the more data a classifier has, the better its performance will be (Sordo & Zeng, 2005). The more data a classifier has to train itself with, the more accurate and comprehensive its understanding of the different classes it will be, and thus the better it will be at distinguishing samples from a one class from those of another. This was most clearly evident in

the results between the runs with all 15 classes (Figure 7) and those with just two (Figure 8). The latter had more samples per class, and therefore the classifier was nearly three times as accurate. It is thus no surprise that the classifier failed entirely to classify the texts from classes with few samples (e.g. humour and science fiction) correctly. There was simply not enough information available to the classifier to allow it to distinguish those samples from those of other classes. However, that was not the only problem that presented itself due to the dataset.

This unequal distribution of samples across the classes had an effect on the precision and recall of the classifier. If a classifier with binary data (i.e. two classes) has a skewed dataset where one class has considerably more samples than the other, the minority class will have more negative samples that can become false positives. Conversely, there are fewer positives that could become false negatives. In other words, it will have low precision, but high recall scores. The reverse would be true for the majority class. Following this logic, an effect along those lines should be affecting the performance of the classifier. The exact effect of a dataset with an unbalanced distribution across 15 classes is unknown, but it is likely detrimental. As such, the Brown Corpus's structure is at least in part the reason for the fairly low performance scores of the classifier. Finally, there was one major self-imposed limitation.

The present study was set up in such a way that the four feature sets were tested solely as whole sets. In other words, the classifier was not run using only individual features. Although doing so would have provided more detailed results, there was one simple reason why it was not done, namely: time. Running the classifier with each possible combination of feature sets took a considerable amount of time. So much so that doing the same for all individual features would have been infeasible. However, in order to still get some insight into this matter, an attempt was made at determining the effect of each individual emotional feature through statistical analyses.

The most interesting of these results was that there was a strong positive and highly significant relation between the effect size of a feature and its number of significantly different class combinations. This meant that the more variance of the samples that a feature can account for, the more useful it is in distinguishing samples from one class from those of another. Particularly useful in that regard were the features: Imagery, Joy, Pleasantness, and Trust. The first is from the Dictionary, and latter three are from the EmoLex. However, these results say very little about why those four features perform better and if the classifier would perform best when using solely those four features. It could be the case that the remaining features of the Dictionary and the EmoLex are in fact detrimental to the performance of those four features. However, it could also be the case that all four of those features help in distinguishing the same classes. For example, they might all be used to distinguish Learned from other classes, but they then can't do the same for other classes. In this case, the four features would not perform any better when used without the other features; they would still have performance issues. As is clearly the case, more study of the individual features is required. That is, however, not the only aspect that warrants further investigation.

Most curious of all results was that there did not appear to be any relation whatsoever between the number of times the post-hoc tests managed to find a significant difference between two classes, and the performance of the classifier based on the values of the confusion matrices. Logically speaking, it stands to reason that the more often that two classes are significantly different from one another, the less frequently the texts from those two classes are classified as false negatives and positives for each other. This did not appear to be the case according to the correlation tests, which suggests that there are other factors in play.

Without further testing, however, it is impossible to point out with certainty what those factors are and what the effect is that those factors have. At the very least, it is likely that genre is a considerable factor, as mentioned previously. The texts were categorised according to an unknown definition of genre during the compilation of the corpus. As a result, it is simply unknown and impossible to determine what the relation is between the texts. Another factor would be sample size. As already mentioned, a balanced dataset, with regard to sample size, would probably have performed better. The greater the sample size, the more information is available about the classes, thus reducing the amount of uncertainty about whether the samples from one class differ from those of another. Sample size thus has an effect on effect size, any significance scores involving the sample size, and the performance of the classifier as a whole. Another probable factor is the influence of features on other features within the same feature set, as also mentioned previously. Although some features might have performed well individually, they could have been held back by other features from the same set. However, there are undoubtedly other factors that influence the performance of the classifier. For instance, the experiment was not run with multiple classifiers. Doing so could have shown that a particular classifier or type of classifier is more suited for this type of study. Another is that the texts in the Brown Corpus were divided into subgenres. For example, each of the fiction genres had two subcategories: short stories and novels. It could be argued that texts from different subgenres are different enough from one another that putting those texts into the same class would have negatively affected the performance of the classifier. It is unlikely, but also impossible to know for certain without further analysis. All told, however, the present study has produced interesting results that provide an initial insight into the potential of the combination of lexical and emotional features to classify texts.

Because the Bag-of-words feature set overpowered the other feature sets, the answer to the present study's research question has to be looked for in the results of the runs that did not include that feature set. When used individually, none of the remaining three feature sets performed particularly well, let alone as well as the Bag-of-words set. When combined, however, they performed better. Of most interest is the combination of the General and Dictionary feature sets, as it had the best performances of the runs with 15 classes. This suggests that a combination between the two types of features could indeed result in a more effective classification approach. The reason for this is likely because the lexical features served to distinguish certain classes, while the emotional features did the same for others. Indeed, when looking at the average F1 scores (Table 7), the Dictionary features provide the necessary values to properly classify texts from, for instance, General Fiction and Adventure Westerns, while the General features do the same for Popular Lore and Miscellaneous. The possible reasons for these results have already been discussed above and, while no certain answers have presented themselves, it is fair to say that these results alone hint at a positive answer to the present study's question.

6 – Conclusion

The present study did not touch upon the concept of recommending texts based on the features of another as described in the Introduction. The reason for this is simple: the approach first had to be explored. Genre is too broad and vague a concept to base accurate recommendations on. As such, genres often merely serve only to indicate the topic of a text, ignoring the use of both technical and affective language. Therefore, basing recommendations on text classification was the logical next step. However, text classifications thus far had only dealt with either lexical or emotional features. The issue with this was that lexical features only deal with technical use of language, while emotional features only show the use of affective language. This suggested that a combination of both types of features would be required to recommend a properly similar text. As such, the present study was set up to serve as a proof of concept. The main focus of this study was to test and describe the possibility and efficacy of using a combination of two emotional and lexical features for text classification. Although both have been used separately for such purposes, they had yet to be combined. Furthermore, the usefulness of the Dictionary of Affect in Language and the EmoLex feature sets for classification purposes was determined.

The results showed that a combination between the two types of features could indeed produce better results than either used separately. The reason for this was in part because the lexical features allowed for distinctions to be made between certain classes, whereas the emotional features did the same for other classes. However, while their performance was among the best, combinations of the two types of features still left much to be desired. Further analysis suggested that not all features of the two emotional feature sets were equally useful for classifying texts. Although the results were limited, they did certainly show that the

demonstrated approach has merit. However, before this approach can be used, additional research should be performed.

Any future research that intends to further explore this approach would do well to be aware of the limitations of the present study. Based on these limitations, there are four recommendations to be made for future studies. Firstly, an alternative to the Bag-of-words approach should be considered. Although this approach performs fairly well when used separately, it is simply not suited to combine with other features. The number of features that the Bag-of-words approach produces is so great that its features overpower those of other approaches. As such, a different lexical feature should be employed. Although no alternative was tested in the present study, results from other studies suggest that features such as pronouns, determiners, negations, and prepositions would serve as suitable replacements. Secondly, the dataset used ought to be balanced and have a sufficient number of samples. Under- and overrepresentation of classes negatively influences the performance of the classifier. Not having enough samples to train the classifier on would be similarly detrimental. Thirdly, any study with a similar purpose would be well served by starting with a large number of features. After running a preliminary round of tests, the least useful features should then be eliminated. Doing so will prevent any less useful features from negatively influencing the performance of other features. Fourthly and finally, there is the matter of genre. Ideally, a dataset would be created specifically for the study. This would make any effects that genre might have on the results known variables. This would in turn be optimal when trying to analyse the usefulness of the approach demonstrated in this thesis. However, this is likely not feasible in many cases. If it indeed is not, try to find and use a dataset for which the criteria that were used to categorise the texts into

genres are described. This should still allow for more accurate and detailed interpretations of the results and the factors that might have influenced them.

The present study was written as a thesis for the MA Digital Humanities. The reason for mentioning this is because the potential value it might have for the disciplinary field is considerable. The field of Digital Humanities is focused around the concept of applying computational methods to any fields that may be improved by doing so. In this particular case, the approach described in the present study could potentially replace the mostly analogue and often arbitrary concept of genre. At the very least, it would enrich it tremendously. The combination of lexical and emotional features encompasses most, if not all aspects of texts. In theory, this should allow it to create classes of texts that are actually alike. This in turn should open up plenty of opportunities of both practical and research natures. The first of which is using the approach to create accurate and useful recommendations. Aside from it being useful as a consumer, it would have valuable commercial applications. It would also allow for studies to determine whether the style of writing differs depending on the gender of the author, their age, their country, etc. Furthermore, it would serve as an indication that the approach regarding films mentioned in the Introduction should be similarly feasible.

References

Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C. D., & Stamatopoulos, P. (2000). *Learning to filter spam e-mail: A comparison of a naive Bayesian and a memory-based approach*. Athens, Greece: National Centre for Scientific Research.

Bayes, T. (1764). An Essay Toward Solving a Problem in the Doctrine of Chances. *Philosophical Transactions of the Royal Society of London, 53*, 370-417. Retrieved from http://www.stat.ucla.edu/history/essay.pdf

Bhatia, V. K. (2004). *Worlds of Written Discourse: A Genre-Based View*. London: Continuum.

Biber, D. (1986). Spoken and Written Textual Dimensions in English: Resolving the Contradictory Findings. *Language, 62*(2), 384-414. doi:10.2307/414678

Biber, D. (1988). *Variations across speech and writing*. Cambridge: Cambridge University of Cambridge.

Brooks, J. A., Shablack, H., Gendron, M., Satpute, A. B., Parrish, M. H., & Lindquist, K. A. (2017). The role of language in the experience and perception of emotion: A neuroimaging meta-analysis. *Social Cognitive and Affective Neuroscience, 12*(2), 169-183. doi:10.1093/scan/nsw121

Ferguson, K. (2017). Digital Surrealism: Visualizing Walt Disney Animation Studios. *Digital Humanities Quarterly 11.1*. http://www.digitalhumanities.org/dhq/vol/11/1/000276/000276.html

Francis, W. N., & Kuçera, H. (1967). A computational analysis of present-day American English. Providence, RI: Brown University Press.

Francis, W. N., & Kuçera, H. (1979). BROWN CORPUS MANUAL. Retrieved from http://www.hit.uib.no/icame/brown/bcm.html#bc10

Geest, D. D., & Gorp, H. V. (1999). Literary genres from a systemic-functionalist perspective. *European Journal of English Studies, 3*(1), 33-50. doi:10.1080/13825579908574428

Gilbers, D. G., Bos, L., Heeres, T., Muller, M., Wierenga, E., & de Vries, N. (2010). *Modaliteit als parameter: verschillen tussen spontane en geacteerde spraak [Modality as parameter: differences between spontaneous and acted speech]*. *TABU, 38(1-4),* 110 - 120.

Haan-Vis, K. D., & Spooren, W. (2016). Informalization in Dutch journalistic subgenres over time. *Genre in Language, Discourse and Cognition,* 137-163. doi:10.1515/9783110469639-007

Heras, D. C. (2012). The Malleable Computer: Software and the Study of the Moving Image. *FARMES Cinema Journal*. Retrieved from http://framescinemajournal.com/article/the-malleable-computer/

Hoffman, P., Brouwer, G., Van Dalfsen, A. (2018) *Week 2 assignment Data as Culture*. Unpublished essay. University of Groningen, Groningen.

Hyland, K. (2002). Genre: Language, Context, And Literacy. *Annual Review of Applied Linguistics, 22*, 113-135. doi:10.1017/s0267190502000065

IBM Corp. Released 2017. IBM SPSS Statistics for Windows, Version 25.0. Armonk, NY: IBM Corp.

Joyce, B., & Deng, J. (2017). Sentiment analysis of tweets for the 2016 US presidential election. *2017 IEEE MIT Undergraduate Research Technology Conference (URTC).* doi:10.1109/urtc.2017.8284176

Kamien, R. (2008). *Music: an Appreciation.* Boston: McGraw-Hill Higher Education.

Karlgren, J. (1999). Stylistic Experiments in Information Retrieval. *Text, Speech and Language Technology Natural Language Information Retrieval*, 147-166. doi:10.1007/978-94-017-2388-6_6

Kessler, B., Numberg, G., & Schütze, H. (1997). Automatic detection of text genre. *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics* -. doi:10.3115/979617.979622

Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment Analysis of Short Informal Texts. *Journal of Artificial Intelligence Research, 50*, 723-762. doi:10.1613/jair.4272

Koppel, M., Argamon, S., & Shimoni, A. (2002). Automatically Categorizing Written Texts by Author Gender. *Literary and Linguistic Computing, 17*(4), 401-412. doi:10.1093/llc/17.4.401

Kotrlik, J., & Williams, H. (2003). The Incorporation of Effect Size in Information Technology, Learning, and Performance Research. Information Techology, Learning, and Performance Journal, 21(1), 1-7. Retrieved July 13, 2018.

Lantz, B. (2015). *Machine Learning with R.* Packt Publishing Limited, 318.

Mohammad, S. M., & Turney, P. D. (2010). Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text* (pp. 26–34). New York, NY: Association for Computational Linguistics.

Mohammad, S. (2012). Portable Features for Classifying Emotional Text, in: the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 587–591.

Mohammad, S.M., & and Turney, P.D. (2013). Crowdsourcing a word–emotion association lexicon. Computational Intelligence, 29(3):436–465.

Murphy, K. R., Myors, B., & Wolach, A. H. (2014). *Statistical power analysis: A simple and general model for traditional and modern hypothesis tests*. London: Routledge.

Ofoghi, B., & Verspoor, K. (2017). Textual Emotion Classification: An Interoperability Study on Cross-Genre Data Sets. *AI 2017: Advances in Artificial Intelligence Lecture Notes in Computer Science,* 262-273. doi:10.1007/978-3-319-63004-5_21

Petrenz, P. (2009). *Assessing Approaches to Genre Classification*. M.Sc. thesis, School of Informatics, University of Edinburgh.

Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A., & Stoyanov, V. (2015). SemEval-2015 Task 10: Sentiment Analysis in Twitter. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. doi:10.18653/v1/s15-2078

Sarlan, A., Nadam, C., & Basri, S. (2014). Twitter sentiment analysis. *Proceedings of the 6th International Conference on Information Technology and Multimedia*. doi:10.1109/icimu.2014.7066632

Schneider, C. A.; Rasband, W. S. & Eliceiri, K. W. (2012), "NIH Image to ImageJ: 25 years of image analysis", Nature methods 9(7): 671-675, PMID 22930834

Schreuder, M., Eerten, L. van & Gilbers, D. (2006). Minor and Major in Emotional Speech. *TABU 35 (1/2)*, 1-15.

Sordo, M., & Zeng, Q. (2005). On Sample Size and Classification Accuracy: A Performance Comparison. *Biological and Medical Data Analysis Lecture Notes in Computer Science*, 193-201. doi:10.1007/11573067_20

Srivastava, A., Singh, M., & Kumar, P. (2014). Supervised Semantic Analysis of Product Reviews Using Weighted k-NN Classifier. *2014 11th International Conference on Information Technology: New Generations*. doi:10.1109/itng.2014.99

Swales, J. M. (2008). *Genre analysis: English in academic and research settings*. Cambridge, U.K.: Cambridge University Press.

Swales, J. M. (2016, January 06). Retrieved September 7, 2018, from https://www.youtube.com/watch?v=W--C4AzvwiU

Whissell, C. (1989) The Dictionary of Affect in Language. In R. Plutchik & H. Kellerman (Eds.), *Emotion: theory, research, and experience. Vol. 4.* New York: Academic Press. Pp. 113-131

Whissell, C. (2007). Quantifying Genre: An Operational Definition Of Tragedy And Comedy Based On Shakespeares Plays. *Psychological Reports, 101*(5), 177. doi:10.2466/pr0.101.5.177-192

Whissell, C. (2009). Using the Revised Dictionary of Affect in Language to Quantify the Emotional Undertones of Samples of Natural Language. *Psychological Reports, 105(2),* 509-521. doi:10.2466/pr0.105.2.509-521

Wolpert, D. H. (1996). The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation, 8*(7), 1341-1390. doi:10.1162/neco.1996.8.7.1341

Yang, Y., & Joachims, T. (2008). Text categorization. Retrieved July 9, 2018, from http://www.scholarpedia.org/article/Text_categorization

Sasaki, Y. (2009). Introduction to Text Classification. Retrieved from https://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/tutorial-TC.html

Zechner, N. (2013). The Past, Present and Future of Text Classification. *2013 European Intelligence and Security Informatics Conference*. doi:10.1109/eisic.2013.61

Appendices

Appendix A - 20 most common words per class

--Press_rep--
said
mrs.
would
new
one
last
two
mr.
first
state
year
president
home
also
made
time
years
three
house
week

--Press_ed--
would
one
new
mr.
united
may
people
american
world
time
first
many
states
state
two
public
us
even
war

years

--Press_rev--
one
mr.
new
music
first
man
well
would
may
time
american
good
great
program
many
could
two
like
jazz
last

--Religion--
god
one
new
world
church
may
would
man
spirit
us
could
christ
also
life
must
many
members

christian
power
even

--Skill_hob--
one
new
may
time
first
many
two
also
good
used
use
make
feed
work
water
must
long
would
much
years

--Pop_lore--
one
would
time
may
new
first
could
people
many
two
even
made
school
years
also
another

must
used
good
still
--Bel_bio_mem--
one
would
new
time
man
even
may
could
first
life
world
two
also
like
men
must
us
well
said
much

--Misc--
state
year
states
may
united
new
development
one
would
made
business
government
1
years
time
must
fiscal
shall
general
tax

--Learned--
af
one
may
would
two
first
1
also
time
must
used
2
system
number
state
made
many
could
even
much

--Gen_fic--
would
said
one
could
like
man
back
time
came
get
little
old
went
know
thought
two
go
men
looked
never

--Mys_det--
said
would
one
back
could
like
man
get
two
know
go
time
got
door
see
went
around
still
right
car

--Sci_fi--
would
could
said
one
time
ekstrohm
helva
mercer
long
like
know
people
hal
b'dikkat
mike
ship
back
man
jack
first

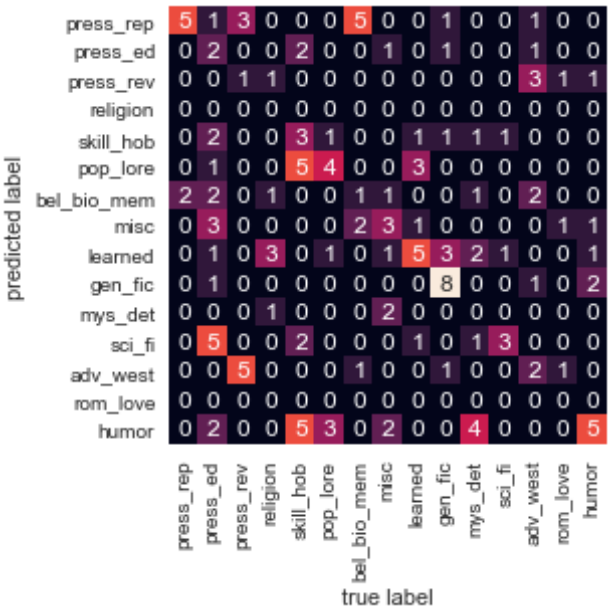
--Adv_west--	--Rom_love--	--Humour--
said	said	said
would	would	one
back	could	would
man	like	time
could	one	even
like	back	line
time	thought	mr.
get	little	could
eyes	man	way
two	get	things
even	time	two
men	old	get
see	got	little
got	know	mother
right	never	arlene
came	even	man
made	way	made
go	went	us
face	go	years
	come	day

Appendix B - List of most common words and for how many classes they were the most common

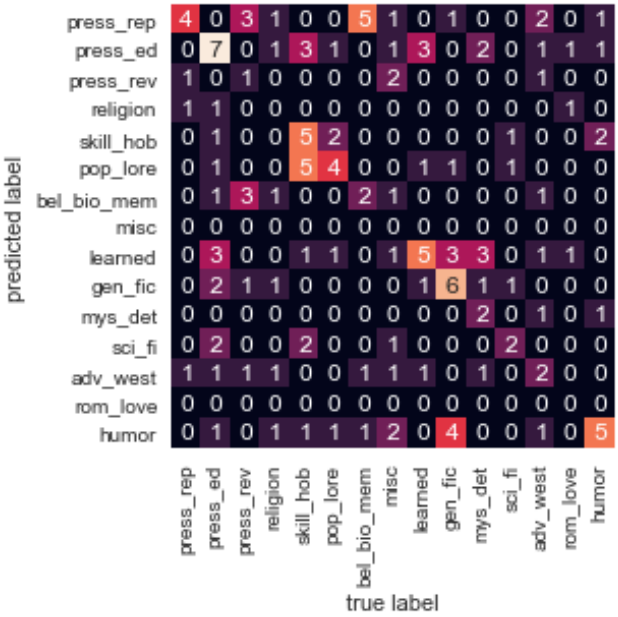
15 would
15 one
14 time
11 two
11 could
9 man
8 said
8 new
8 may
8 first
8 even
7 like
6 years
6 must
6 many
6 made
6 also
5 get
5 back
4 us
4 state
4 mr.
4 know
4 go
3 world
3 went
3 used
3 people
3 much
3 men
3 little
3 got
3 good
2 year
2 well
2 way
2 united
2 thought
2 still
2 states
2 see
2 right
2 old
2 never
2 long
2 life
2 last
2 came
2 american
2 1
1 work
1 week
1 water
1 war
1 use
1 three
1 things
1 tax
1 system
1 spirit
1 ship
1 shall
1 school
1 public
1 program
1 president
1 power
1 number
1 music
1 mrs.
1 mother
1 mike
1 mercer
1 members
1 make
1 looked
1 line
1 jazz
1 jack
1 house
1 home
1 helva
1 hal
1 great
1 government
1 god
1 general
1 fiscal
1 feed
1 face
1 eyes
1 ekstrohm
1 door
1 development
1 day
1 come
1 church
1 christian
1 christ
1 car
1 business
1 b'dikkat
1 around
1 arlene
1 another
1 af
1 2

Appendix C

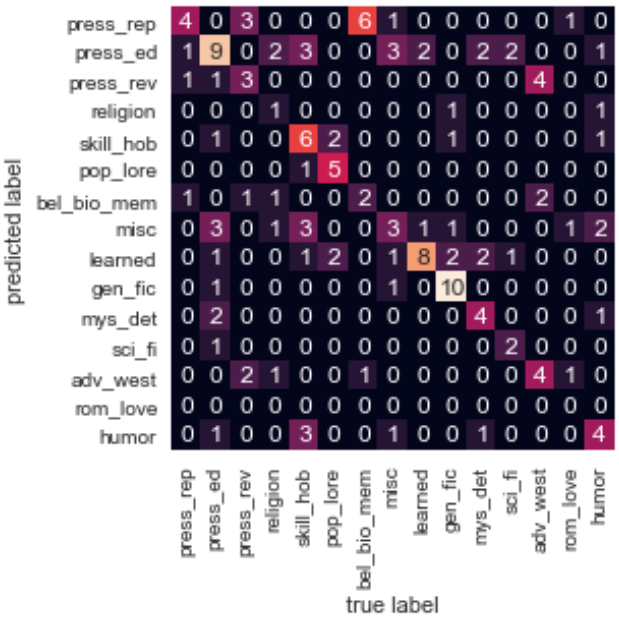
GassuianNB – All words



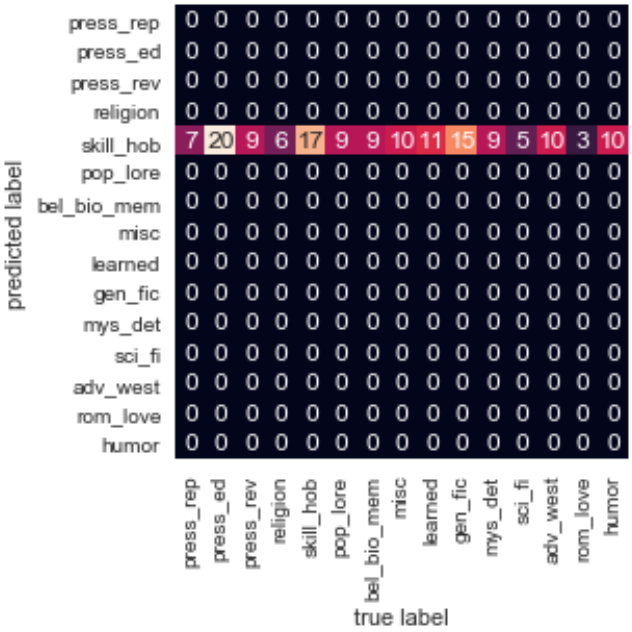
All emotional features



MultiNomialNB – All words



All emotional features



Appendix D - Post-hoc results; number of significantly different features between indicated classes

Class	Compared with	Dictionary (out of 3)	EmoLex (out of 10)	Combined (out of 13)		Total
Press_rep	Adv_west		6	6		41
	Bel_bio_mem	1	3	4		
	Gen_fic		5	5		
	Humour					
	Learned	3		3		
	Misc	1	2	3		
	Mys_ det		4	4		
	Pop_lore		1	1		
	Press_ed	1	1	2		
	Press_rev	1	2	3		
	Religion	1	5	6		
	Rom_love		6	3		
	Sci_fi					
	Skill_hob		1	1		
Press_ed	Adv_west	1	2	3		31
	Bel_bio_mem					
	Gen_fic	1	2	3		
	Humour					
	Learned	1	5	6		
	Misc		4	4		
	Mys_ det	1	2	3		
	Pop_lore					
	Press_rep	1	1	2		
	Press_rev	1		1		
	Religion		1	1		
	Rom_love		2	2		
	Sci_fi					
	Skill_hob	2	4	6		
Press_rev	Adv_west	2	3	5		34
	Bel_bio_mem					
	Gen_fic		1	1		
	Humour					
	Learned	3	2	5		
	Misc		4	4		
	Mys_ det	2	2	4		
	Pop_lore	1	1	2		
	Press_ed	1		1		
	Press_rep	1	2	3		
	Religion		3	3		
	Rom_love		1	1		
	Sci_fi					
	Skill_hob	2	3	5		
Religion	Adv_west	2	4	6		72
	Bel_bio_mem		4	4		
	Gen_fic	1	3	4		
	Humour		2	2		

	Learned	2	8	10		
	Misc		8	8		
	Mys_ det	3	4	7		
	Pop_lore	1	4	5		
	Press_ed		1	1		
	Press_rep	1	6	7		
	Press_rev		4	4		
	Rom_love	1	3	3		
	Sci_fi		2	2		
	Skill_hob		9	9		
Skill_hob	Adv_west		6	6		56
	Bel_bio_mem	1	5	6		
	Gen_fic		3	3		
	Humour		1	1		
	Learned	2		2		
	Misc	1	3	4		
	Mys_ det		5	5		
	Pop_lore		5	5		
	Press_ed	2	4	6		
	Press_rep					
	Press_rev	2	3	5		
	Religion	1	9	10		
	Rom_love		3	3		
	Sci_fi					
Pop_lore	Adv_west	1	2	3		33
	Bel_bio_mem	1		1		
	Gen_fic	1	2	3		
	Humour					
	Learned	3	1	4		
	Misc	1	6	7		
	Mys_ det		2	2		
	Press_ed					
	Press_rep		1	1		
	Press_rev	1	1	2		
	Religion	1	4	5		
	Rom_love		1	1		
	Sci_fi					
	Skill_hob		4	4		
Bel_bio_mem	Adv_west	2	3	5		44
	Gen_fic	1	2	3		
	Humour					
	Learned	1	7	8		
	Misc		6	6		
	Mys_ det	2	2	4		
	Pop_lore	1		1		
	Press_ed					
	Press_rep	1	3	4		
	Press_rev					
	Religion		4	4		
	Rom_love	1	2	3		
	Sci_fi					

	Skill_hob	1	5	6		
Misc	Adv_west	1	7	8		72
	Bel_bio_mem					
	Gen_fic	1	7	8		
	Humour		4	4		
	Learned	2	2	4		
	Mys_ det	1	6	7		
	Pop_lore	1	6	7		
	Press_ed		4	4		
	Press_rep	1	2	3		
	Press_rev	1	5	6		
	Religion		8	8		
	Rom_love	1	7	8		
	Sci_fi		1	1		
	Skill_hob	1	3	4		
Learned	Adv_west	1	7	8		67
	Bel_bio_mem	2	7	9		
	Gen_fic	2	5	7		
	Humour	1		1		
	Misc	2	2	4		
	Mys_ det	1	2	3		
	Pop_lore	3	1	2		
	Press_ed	1	5	6		
	Press_rep	3		3		
	Press_rev	3	2	5		
	Religion	2	8	10		
	Rom_love	2	5	7		
	Sci_fi					
	Skill_hob	2		2		
Gen_fic	Adv_west					37
	Bel_bio_mem	1	2	3		
	Humour					
	Learned	2	5	7		
	Misc	1	7	8		
	Mys_ det					
	Pop_lore	1	2	3		
	Press_ed	1	2	3		
	Press_rep		5	5		
	Press_rev	1	1	2		
	Religion	1	3	4		
	Rom_love					
	Sci_fi					
	Skill_hob		2	2		
Mys_det	Adv_west					40
	Bel_bio_mem	2	2	4		
	Gen_fic					
	Humour					
	Learned	1	2	3		
	Misc	1	6	7		
	Pop_lore		2	2		

	Press_ed	1	2	3		
	Press_rep		4	4		
	Press_rev	1	2	3		
	Religion	3	4	7		
	Rom_love	1	1	2		
	Sci_fi					
	Skill_hob		5	5		
Sci_fi	Adv_west	1		1		4
	Bel_bio_mem					
	Gen_fic					
	Humour					
	Learned					
	Misc		1	1		
	Mys_ det					
	Pop_lore					
	Press_ed					
	Press_rep					
	Press_rev					
	Religion		2	2		
	Rom_love					
	Skill_hob					
Adv_west	Bel_bio_mem	2	3	5		52
	Gen_fic					
	Humour					
	Learned	1	7	8		
	Misc	1	7	8		
	Mys_ det					
	Pop_lore	1	2	3		
	Press_ed	1	2	3		
	Press_rep		6	6		
	Press_rev	1	3	4		
	Religion	2	4	6		
	Rom_love	1	1	2		
	Sci_fi	1		1		
	Skill_hob		6	6		
Rom_love	Adv_west	1	1	2		40
	Bel_bio_mem	1	2	3		
	Gen_fic					
	Humour					
	Learned	2	5	7		
	Misc	1	8	9		
	Mys_ det	1	1	2		
	Pop_lore		1	1		
	Press_ed		2	2		
	Press_rep		6	6		
	Press_rev		1	1		
	Religion	1	3	4		
	Sci_fi					
	Skill_hob		3	3		
Humour	Adv_west					6

	Bel_bio_mem					
	Gen_fic					
	Learned	1		1		
	Misc		3	3		
	Mys_ det					
	Pop_lore					
	Press_ed					
	Press_rep					
	Press_rev					
	Religion		1	1		
	Rom_love					
	Sci_fi					
	Skill_hob		1	1		