POSTGRADUATE THESIS

Vougioukas Evangelos Alexandros

MBADS Registration Number: 19009

Screenplay analysis for the detection of racist/homophobic content

Supervisor: Kaparis Konstantinos

Assistant Professor

Department of Business Administration

University of Macedonia

January 4th, 2022

Contents

Table of Figures

Table of Tables

Glossary

SA	**Sentiment Analysis**
NLP	**Natural Language Processing**
NB	**Naïve Bayes**
RNN	**Recurrent Neural Networks**
SVM	**Support Vector Machine**
CBOW	**Continuous Bag of Words**
OCC	**Ortony, Clore, and Collins**
BoW	**Bag of Words**
CNN	**Convolutional Neural Network**
ML	**Machine Learning**
AI	**Artificial Intelligence**
IMSDb	**Internet Movie Script Database**
URL	**Uniform Resource Locator**
DNN	**Deep Neural Network**
NBM	**Naïve Bayes Model**
GNB	**Gaussian Naive Bayes**
MNB	**Multinomial Naive Bayes**

Abstract

Some of the worst traits of human beings are racism and homophobia and they get even worse when they are expressed through art. These problematic behavioral incidents mainly appear throughout movies and plays. There is a lot of discussion around the world concerning these incidents both online and offline. Some groups consider that developed countries, such as the USA, the UK and so forth, are not free from racism and others view it is as misinformation processing. Therefore, there is a requirement for a system to evaluate and determine the possible scenarios regarding racism worldwide. In the present thesis, sentiment analysis will be used in order to identify racist and homophobic content in movies and plays.

1 Introduction

1.1 Background

As human beings, we have many demeaning feelings and behaviors indicative of a predisposal for negative comportment. Some of these feelings and behaviors have even existed for many decades or even centuries. Furthermore, proponents of these feelings and behaviors try to convince other people that these are correct and should be advocated. Two of the most characteristic behaviors are racism and homophobia. Racism is the notion according to which people are not equal, but are divided into superior and inferior individuals, distinguished by skin color, ethnicity, religion, gender or sexual orientation. The most common type of racism, and what has given the word its original name (race/razza= race), is racial racism. Homophobia is the aversion or discrimination based on sexual orientation or more generally to the detriment of LGBTQI+ individuals. The word is etymologically derived from the Greek words homo (like) and phobia. It was first used in 1969 by the American clinical psychologist George Weinberg who became known for his book Society and the Healthy Homosexual (1971). As a broader concept, it refers to all LGBTQI+ individuals and not necessarily exclusively homosexuals (Wikipedia, 2021).

Nowadays, everyone can express their beliefs and there are many public or private means that can be used in order to influence people. Movies or plays are among the most common means that the majority of people use when they want to entertain themselves. Apparently, through this implies that anyone can express their beliefs to a broader and sometimes global audience. So far, there have been reported many cases where illegal or at least problematic opinions existed in movies or plays. In some cases, these productions were cancelled, or in some others the scenario, the dialogues and so forth have been changed accordingly. Unfortunately, there has also been misinformation regarding certain productions without problematic elements. Contemporary technology has dramatically evolved offering us plenty of tools and solutions. An example of those could be sentiment analysis that can help us identify "problematic" content within scripts of movies or plays. Sentiment analysis or opinion mining, is the process of using natural language processing, text analysis and statistics to analyze opinions, sentiments, evaluations, attitudes, and emotions from individuals towards entities such as products, services, events, topics and their attributes. It mainly focuses on language on which positive or negative sentiments are expressed or implied. While sentiment analysis is a sub-task of artificial intelligence (or more strictly, the processing of natural language), it remains rather vaguely defined. This is supported by levels of agreement among human commentators, which are valued at around 80%. This can be seen in some ways as a target level for sentiment detection accuracy, starting from further improvements required in order to achieve a more detailed algorithm setup. In part, the complexity of the job definition refers to how each human commentator perceives a sentiment that is demonstrated through an expression depending on their current mood, personal attitude towards the subject of expression and the purpose of the commentary. Rule-based algorithms produce a good level of clarity that increases the ability to answer the simple question of why the system classifies specific sentiment. (Kan, 2012), (Liu, 2012)

1.2 Thesis Development

Nowadays, movies and plays are produced at a growing rate. Some of those may contain content that is disputable. On the other hand, an individual or a group of people may accuse a production of promoting illegal content either due to ignorance or due to their very own purposes. Thanks to the evolution of technology, there are many techniques that can help authorities to find illegal content in productions. Two common forms of illegal content are racist and homophobic content. The discovery of such content is not censorship, as someone might say, but prevention against illegal or unethical productions that may lead to unwanted situations. The most well-known and widespread methodology for analyzing movie and play scripts is sentiment analysis.

In the present thesis, the topic of sentiment analysis in movies and plays is going to be dealt with, with the purpose of discovering racist and homophobic content. To begin with , a definition of what racism and homophobia are will be given . In addition, illegal content, including racist and homophobic content, in movies and plays will be examined. Some samples examining how local authorities and community dealt with such content will be presented and sentiment analysis and machine learning will be described, as well as how they are applied in movie scripts and plays. Furthermore, a case study will be implemented in python programming language detecting racist and homophobic content in movie or play scripts. The results of the case study will be evaluated and, finally, conclusions will be drawn based on this specific work, while some suggestions for future work will be made.

1.3 Structure

Throughout the first section, the background, the methodology and structure of the thesis will be introduced. The second section contains the literature review divided into five sub-sections dealing with racism and homophobia definitions, illegal content in movies and plays in general, sentiment analysis, machine learning and sentiment analysis regarding movie and play scripts. In the third section, the case study as well as the implementation in Python will be presented. The fourth section contains the evaluation results derived from the case study implemented in the previous section. Finally, conclusions and suggestions for future work are provided in the fifth section.

2 Literature Review

2.1 Definition of Racism and Homophobia

2.1.1 Racism

The term racism involves hatred or fear of people belonging to races other than ours, as well as having a hostile or even derogatory attitude towards them, and even demonstrating systematic restrictions and discrimination against them. This behavior relates to the belief that these people belong to races that are inferior to ours. The term racism is consequently used for similar perceptions and attitudes towards individuals different from ourselves, in relation to other distinctive features besides race, such as gender, ethnic origin, social status, state of health and so forth.

The concept of race is linked to the concept of racism. The former refers to a group of people who are of common descent and usually have certain distinct physical characteristics, such as skin color. This concept is disputed, especially after DNA having been decoded, as racial markers have been shown to constitute a minimal, negligible percentage of an individual's genetic makeup. Nowadays, social scientists agree that race is a social construct rather than a genuine biological category. All people have a common origin and the groups belonging to the single human species have been constantly migrating, resulting in admixtures. Thus, racial differences are relative and not absolute.

Many people agree that racism emerged as a scientific theory. The term biological racism refers to theories that supported racial inequality based on biological arguments and were linked to social Darwinism (the indistinguishable transfer of Darwin's biological theory to society), according to which the strongest ones in society have the right to survive. Gobineau was considered to be the founder of racial determinism, which gave the western world the necessary justification for its colonial and imperialist plans. At its peak, racial theory and practice reached Nazism.

Racism or racialism has often relied on various scientific theories. In 1994, researchers Herrnstein and Murray reiterated the claim that there was a correlation between race and intelligence, arguing that blacks were inferior because they performed worse in certain intelligence tests, to which white and black children were subjected. However, those tests have been criticized by many other scientists as having been designed from the point of view of middle-class white people, meaning that the questions favored children who are the offspring of this category, while leading to poor performance when it came to children of other populations. It is argued that intelligence is not distributed according to race but arises from a combination of genetic and environmental factors, especially the educational and economic status of families. Most researchers worldwide consider the following factors as crucial for children's mental performance:

- different opportunities and stimuli provided to children from infancy, when they grow up in environments of different financial and educational level,
- different kind of help they receive during their school years and
- different parental expectations concerning their future.

Thus, the higher economic and educational level that white people had for historical reasons has been reproduced in subsequent generations. The experiment repeated in several classes by the American teacher Jane Elliot (see http://www.youtube.com/watch?v=Bf0zfMl5KSA) proved that school performance is a function of both self-esteem and self-confidence that the attitude of teachers and classmates towards students cultivates.

The term racialism was formerly used as a synonym for racism. Nowadays, racialism is used as a seemingly neutral acceptance of the existence and distinction of races, implying modern racism that focuses on the cultural and social differences of "races" and not the biological ones, which the modern science of biology has rejected. In other words, contemporary perception of racism has changed so much that we are now talking about "racism without races". According to Pierre-André Taguieff, this is thought to be "differential racism".

According to Andreas Pantazopoulos, the ideology of today's new racism, in contrast to the old biological one, can be analyzed into the following four main characteristics, according to Taguieff (The Hellenic Parliament Foundation, 2018):

a) The shift of racist discourse from "race" to "culture". That is, replacing the "argument" for the so-called racial purity with that of defending an "authentic" cultural identity.
b) The shift from "inequality" to "difference". The old devaluation of the "inferior" ("races") today gives way to the phobia of mixing cultures.

c) In resorting to “heterophile” (friendship to the other, the foreigner) formulas (“right to dispute”), rather than to “heterophobes” (fear of the foreigner).
d) The last characteristic could be expressed through the formulation of “indirect” or “symbolic racism”: neo-racist discourse is difficult to conceive as “racist”, because it throws all its weight on the implicit, the implied. Anti-Semites today present themselves as “anti-Zionists”. Anti-immigrant discourse substantiates, universalizes and demonizes the “unwanted” foreigners (“the Arabs”). Thus, the anti-immigrant attitude is presented as a smooth reaction of a society, which considers itself in a state of legal defense against an “invasion”: the invocation of “national preference” (“national citizens first”) is the “logical” culmination of an indirect but powerful policy of exclusion from the national-social body of the “others”.

Biological racialism was associated very early with eugenics, the claimed “science” of improving the human species through selective reproduction. Eugenics is divided into positive and negative. The former encourages the reproduction of supposedly superior human beings. Hitler's Nazi regime had embarked on both a positive eugenics program (blonde, blue-eyed Norwegian women gave birth to blond-born Germans) and a negative eugenics program (sterilized disabled, mentally ill, etc.). An extreme version of a negative eugenics program was the plan to exterminate the Jews, the so-called “Final Solution”, which aimed to purge the human race from the lower Jewish race, to eradicate its “meager” genes.

The Genocide of the Jews (also known as the Holocaust) was the culmination of the discrimination that began in 1935 with the “Nuremberg Laws”, which imposed severe restrictions on the rights of German Jews. In 1938, the “Night of the Crystals” heralded the impending escalation of the Nazi plan, as that night not only were Jewish shops, homes and synagogues vandalized, but also Jews were beaten and killed. In 1939, the systematic planning for the genocide of Jews living in Germany and in the countries occupied by the Nazis was launched. Concentration camps, such as Dachau, which had been set up in Germany since 1933 to isolate the regime's political opponents (communists and trade unionists) and exploit them as laborers, later developed into mass extermination camps for both Jews, Russians and homosexuals. Many more extermination camps inside and outside Germany (Auschwitz, Treblinka, etc.) were added to those concentration camps. Six million Jews were executed shortly after their arrival (especially children, the elderly and the vast majority of women), died of exhaustion, suffered forced labor with little food, or were executed by death squads (Einsatzgruppen) in the Eastern Front.

The peculiarity of the Holocaust (Shoah), according to Enzo Traverso lies in the fact that death, a mechanism created by a minority of crime architects and operated by a mass of sometimes ardent and other times unconscious executors, amid the tacit indifference of the vast majority of the German population, with the complicity of Europe and its "passivity".

Other terms related to racism are xenophobia, apartheid, anti-Semitism, sexism, feminism and heterosexism:

- Xenophobia is the dislike of individuals or groups who are considered to be strangers. Dislike can range from controlled repulsion to pathological phobia.
- Apartheid means "separation" in the Dutch dialect of South Africa. The term refers to the system of discrimination, which institutionalized the superior position of white people in all domains (economic, social and political) and ensured the domination of whites over blacks in South Africa. This set of racist laws was opposed by the "South African Congress of Trade Unions", which, after decades of struggle, succeeded in freeing its leader, Nelson Mandela, in 1990, and then abolishing apartheid.
- Anti-Semitism is called racism against the Jews, which is a subcategory of racial discrimination. The term is actually abusive, as Semites are those who speak Semitic languages, not only the Jews, but also the Assyro-Babylonians, the Phoenicians and the Arabs. The history of anti-Semitism is very old. When Christianity became the dominant religion and later the main force of cohesion in Europe, the Jews were marginalized from society, as they were accused of being responsible for the death of Christ. Of course, those accusations concealed the fact that Christ himself, Virgin Mary and all the Apostles were Jews. Since then, they have been isolated in ghettos (compulsory Jewish settlements) in most European cities. Due to the restrictions placed upon them in terms of access to various professions and public life, combined with the insecurity caused by the hostile climate towards them, they were discouraged from investing in land (and generally in real estate that would be lost in case of forced relocation) and many

turned to commercial and banking activities. The rise of trade at the end of the Middle Ages and during the Renaissance favored many Jews, who, due to the aforementioned occupations, became rich and envied. Thus, discrimination against Jews was intensified and often escalated into persecution, ranging from forced relocation to known pogroms (destruction of property, looting, and murder).

It should be noted that while anti-Semitism is directed against Jews for racial reasons, traditional anti-Semitism, which is apparently the predecessor of anti-Semitism, is directed against them for religious reasons. Western anti-Jewish discourse was made up of two statutory myths:

a) the myth of the ritual murder and the desecration of the holy bread, which created the negative stereotype of the “bloodthirsty Jew”;
b) the myth of the global conspiracy of Judaism-Zionism, which created the inherent negative stereotypes of the “conspirator/sovereign”, the "devilish machinist" and the “naturally deceitful liar”.

The hostility against the Jews, which gradually turned into anti-Semitism, provoked pogroms against them, especially in Eastern Europe, and less intense rivalry in many other countries. In Greece, there were some anti-Jewish events, such as the “Pacific Case” in 1849, the “Slander of Blood” in Corfu in 1891 and the arson of the Campbell settlement in Thessaloniki in 1931. During the interwar period, anti-Semitism was consolidated and spread more widely with the rise of the radical Right and the collapse of economic liberalism and parliamentarism. Establishment of totalitarian regimes in Europe turned anti-Semitism into a formal ideology.

- Sexism is considered as the discrimination among people because of their gender (in English gender = sex), usually meaning racism against women. It is due to a deeply rooted and often unconscious set of beliefs and mentalities which are summed up in the inherent inferiority of people who belong to the female biological sex. This mentality leads to degrading attitudes towards women and their exclusion from those areas of activity that are deemed important (political life, higher education, etc.). The aggressive form of sexism, “male chauvinism”, treats women as objects of male sexual pleasure. A reaction to sexism is feminism.

Feminism constitutes the defense of women's rights and includes the mobilization for political and legal rights, equal opportunities in education and work, sexual autonomy and the right to self-determination, i.e., the free choice of whether and when a woman will become a mother.

- Heterosexism: the term refers to the hostility of heterosexuals towards homosexuals and the discrimination that results from this hostility towards people because of their sexual orientation.

2.1.2 Homophobia

Homosexuality has been found in all cultures since ancient times. However, until a few decades ago, it was considered unacceptable and even a criminal activity in many Western countries. People with a homosexual or bisexual orientation have been stigmatized for a long time. Things changed with the rise of the homosexual political movement in the late 1960s, as well as the removal of homosexuality from the American Psychiatric Association (DSM) diagnostic manual, thus putting an end to the identification of heterosexual behavior as the sole acceptable sexual orientation.

However, there are still many people who continue to consider homosexuality as a mental illness and characterize homosexual individuals as “sick”. The term “homophobia” was coined in 1969 by heterosexual clinical psychologist George Weinberg. According to him, homophobia refers to hatred and intolerance towards homosexual men and women and the irrational fear of close contact with them.

It is associated with a set of thoughts, ideologies, and even actions that oppose sexual orientation, lifestyle, and the rights claimed by homosexuals. It could be argued that it is a generalized type of phobic reaction towards people who have a different orientation from homophobic individuals or those who perceive themselves as “normal”. Homophobia can be expressed in many ways, from discrimination in the workplace, negative media commentary to violent reactions against homosexuals and even hate crimes.

It is then that the following question arises: “Is homophobia a mental illness after all”? The results of many studies conducted so far are contradictory. Some studies suggest that homophobes do not necessarily have a heterosexual orientation. In other words, homophobia can be detected in people with latent homosexual tendencies. In other words, the homophobic person is essentially coming against their own desires. Further research contradicts these findings and argues that factors such as morality, religion, social and family context in which a person is raised can significantly influence views and perceptions regarding homosexuality.

Equally interesting are the findings of a study conducted in 2015, in which a high correlation between people with homophobic behavior and those with psychotic characteristics was detected. Of course, this does not mean that all homophobes have psychological problems, but there are some psychotic characteristics in their personality that range from intense anger to aggression. Recently, surveys have been conducted in schools in order to determine the attitudes and behaviors of students. A fairly recent survey conducted in Madrid demonstrated that school violence is one of the most common problems faced by students. As many as 60% of the students stated that they had witnessed verbal violence against a classmate in relation to their sexual orientation.

It should be noted that children copy and imitate parental behaviors. Thus, when a teenager grows up in a homophobic environment, for example, they are more likely to end up being homophobic. Despite the fact that homosexuality is no longer a taboo for human society –always referring to countries of the western world– prejudice against homosexuals continues to exist in the wider social context. Homophobia is still distinct despite the apparent acceptance of homosexual and bisexual orientation. Let us hope that with the right education and the necessary sex education, the rates of homophobic individuals will be significantly reduced in the future, if not eliminated (Askitis, 2017).

2.2 Racist, Homophobic and Other Illegal Content in Movies and Plays

2.2.1 Images of Gender Race Age and Sexual Orientation

Nowadays, it is more than clear that children's behavior and beliefs are mainly influenced by media portrayal. What has been noticed among various types of research conducted during the previous years is that many stereotypes based on gender, race and sexual orientation have been portrayed in media (Towbin, Haddock, Zimmerman, Lund, & Tanner, 2003).

2.2.1.1 Gender Portrayal

Television has invaded our lives since the early 1950s. Throughout these years, many researchers have found that men are extensively presented compared to women. Further research has led to the conclusion that since the 1950s approximately one one-third of the characters on television have been female. It is more likely for women to be portrayed as thinner and younger compared to men. A characteristic example is that the majority of females on nighttime television programs are young and attractive. What is quite common is the fact that most female characters are either under 35 or over 50. Nowadays, it is common to see middle-aged women on nighttime television programs. Another common situation is that women are not presented as being very clever with a strong, competent personality. Throughout recent years, it has become evident that gender stereotypes abound. Men are presented as being strong, however ineptly handling children's needs, whereas women are objectified more frequently than men.

The main characteristics of men appearing on television are power, violence, tolerance, competition, smartness, rationalism and stability. Among these characteristics, men on television have the tendency to be violent. As many researchers have pointed out, 61 % of television programs contain some kind of violence. It has been found that 44 % of the perpetrators were attractive and in 70 % of the cases they were not punished for the crimes they had committed. Another common aspect is the fact that men are violent against women and this is often portrayed as a form of "heterosexually-based sensuality". In such images, women are often seduced by brutal and abusive men (Towbin, Haddock, Zimmerman, Lund, & Tanner, 2003).

2.2.1.2 Portrayal of Marginalized Racial Groups

Until a few years ago, television portrayal of families mostly involved white and middle-class individuals. Although nowadays more people of color appear on television, Nelson argued that different and accurate depictions of these characters or cultures are rarely provided. Nelson made the following statement about black comedies: "Black comedies are not black in the sense that they present an African-American worldview or black philosophy of life. Rather, they are black because the performers are black".

The most common roles for black actors in the decades of 1930 and 1945 were those of slaves or servants. Issues related to racial discrimination began to emerge after World

War II. Around the 1970s, “blaxploitation” action movies emerged, in which the white “bad guys” were defeated by black heroes. Nowadays, it is very common for white and black actors to be working together on a film. Another thing that is mentioned by Artz is that the friendly genre of films brings out a “new racism”. While the living and working conditions of poor blacks are systematically ignored, some of them appear in roles of successful middle-class people. Artz believes such images help to create a perception of harmonious racial relations.

Another race that did not have much presence in media was Asians. Currently, this seems to have changed and many cultures from Asia having different traditions and lifestyles have been merged into one group. Men from Asia tend to be portrayed either as asexual, non-male males, or experts in martial arts. On the other hand, women from Asia tend to be portrayed as exotic and very attractive women. Mahdzan and Ziegler conducted an informal analysis of film content and found that Asian men are frequently portrayed as inappropriate companions for women of their own race. This is clearly stated in films where both white men and Asian women star. The underlying message of these films is that women from Asia prefer to be in a relationship with white men. Further examining a plethora of films, Mahdzan and Ziegler concluded that scenes of white men having sex with Asian women are more frequent than those of Asian men having sex with white women. They also pointed out that the winners in the battles between Asian and white men were in most cases the latter. Finally, they concluded that white men were portrayed as being stronger and smarter than men from Asia in various movies.

Another marginalized race in terms of projection in films and generally media are Latinos. Latinos are often portrayed as stereotypical, simple or negative models of individuals. For approximately three decades, from the middle ’50s to the middle ’80s, the percentage of television coverage of Latinos was more or less about 2 %. This trend continued up to the middle ’90s. Although the Latino population of the USA increased by 100 %, this low level of coverage continued to exist. A very frequent image of Latinos was that of robbing whites. This changed a little and Latinos started being portrayed as “Latin lovers”. Despite the fact that many stories of “hot romance” existed in films, Latinos never succeeded as interracial lovers. In most recent movies, Latinos are frequently portrayed either as too violent characters or too rebellious (Towbin, Haddock, Zimmerman, Lund, & Tanner, 2003).

2.2.1.3 *Portrayal of older adults*

Signorielli and Bacue found a common practice on television that is the celebration of youth, especially regarding women. On television, it is very common for men to be more or less four years older than women. In the ’90s, the percentage of women that were defined as young adults was higher than in the ’70s and ’80s. Additional research concluded that a minor percentage of the women shown on television was over 50 years old. According to this research, there were many negative messages related to ageing, especially for women. It was more likely for men around the age of 65 to be reported having a job and being classified as middle-aged. On the other hand, women of the same age were portrayed as being older and having to work only within the household (Towbin, Haddock, Zimmerman, Lund, & Tanner, 2003).

2.2.1.4 *Portrayal of gay, lesbian, and bisexual people*

Homosexuals and lesbians accounted for 2 % of the characters in the television season of 1999 – 2000 and most of them played supporting roles. Although the representations of gay characters are increasing, these depictions involve almost exclusively white and male individuals. Colored lesbians and homosexuals remain largely invisible on television.

In the very first years of film production, depictions of same-sex sexuality were banned, and so gay characters were not tender or outspoken, but rather expressed their sexuality through heterosexual behavior. Only in the 1950s and 1960s were there some gay representations in movies, but gay characters were usually sad, alone, and eventually died. According to a negative stereotype that continues to exist when making films, but not statistically proven, homosexual characters are disproportionately portrayed as psychotic and murderous. However, Gross stated that some small steps have been taken for homosexual depictions in the ’90s (Towbin, Haddock, Zimmerman, Lund, & Tanner, 2003).

2.2.2 Retrospective on Classic Movies

In September 2020, a French film starring eleven-year-old girls who dance “provocatively”, wearing shorts and making their pre-adolescent revolution, sparks a wave of reactions on social media worldwide. Angry voices urge Netflix to withdraw it. The “moral war” that broke out referred to the sexualization of young girls, the promotion of toxic patterns, and even the attraction of an audience with dangerous pedophile tendencies. Director Maïmouna Doucouré, a Frenchwoman of color, rushed to defend her work, arguing that she actually wanted to criticize everything she was accused of. The message of the film is deeply feministic and, in its way, scathes current Western standards (Classic films, which today would have been condemned by the public, for messages of racism, sexism and pedophilia, n.d.).

However, distances are narrowed and “moral wars” on the altar of political correctness are unquestionably a sign of our times. Therefore, Doucouré could not escape the trend. Until recently, racism, sexism, and homophobia have remained largely within the bounds of a more academic and theoretical framework. They currently shape public discourse and influence criticism in the workplace, everyday life and art. Thus, it is difficult to imagine famous films of the past being shot based on current data.

2.2.2.1 Lolita (1962, 1997)

Based on the homonymous Russian-American novel by Vladimir Nabokov, “Lolita” revolves around the “forbidden” love of a middle-aged man with his partner's fourteen-year-old daughter. Both the book and the two films based on it present the case from that point of view. As a result, the relationship that develops between them is presented as consensual, although the young protagonist is not at an age where she could consent. It is one of the most controversial literary works of classical literature which had provoked many reactions at the time. However, those that proclaimed freedom in art overshadowed the voices that spoke of “moral legitimacy” of pedophilia. On the other hand, the vast majority of modern review shows that the occasionally written articles about the book or both films tend to be critical. British journalist and writer Rachel Johnson in *Spectator* published one of the most recent and characteristic articles.

Figure 1, Scene from Adrian Lynn's Lolita

“Nowadays, publishers are so terrified of the Twitter crowd that the 'Lolita' handwriting would end up in the closets or in the trash. If the book were in the curriculum of university literature today (and this is a big “if”), it would certainly automatically trigger the urgent warning that this case concerns the "systematic rape of a little girl”. What publisher would dare to publish the book of a middle-aged (or old) white man about a middle-aged (or old) white man who punches a 12-year-old girl? She even pointed out that when she tried to type “Lolita” on Google, the warning immediately appeared “Warning: child pornography is illegal”.

2.2.2.2 American Beauty (1999)

Objections are similar regarding Sam Mendes' *American Beauty*, starring Kevin Spacey. The main character is a 40-year-old well-to-do family man, who is going through a midlife crisis and is sexually attracted to a classmate of his daughter. Gradually his whole life revolves around his fantasies about her. Twenty years ago, *American Beauty* was sweeping the Oscars and it was the film with the best reviews of the year. It was considered to be a modern satire of the “American dream”, starring an antihero who was confronted by his passions.

Figure 2, The young protagonist of American Beauty in one of the most famous scenes

Today, for many it is on the list of movies that cause embarrassment as well as rage. In the context of the #MeToo movement, the Hollywood pedophile scandals and the feminist frenzy, American Beauty is believed to have contributed to the acceptance of the toxic and dangerous patterns that society is struggling to eliminate.

2.2.2.3 Blazing Saddles (1974)

In its release year, *Blazing Saddlers* was a subversive comedy that satirized Americans' favorite genre: the Western. In today's context, the film would probably not even be described as a comedy, as many of the episodes seem more offensive than funny. Characteristically, white protagonists do not hesitate to use the word nigger, which has stigmatized blacks and its use today has been reduced to a serious moral offense. In another scene, one of the characters, in order to prove that he is worth joining a gang, repeatedly emphasizes how much he likes to rape. The way he says is comical and after his words, everyone laughs.

2.2.2.4 Greek Cinema

Offensive jokes and “problematic” cases are not the exclusive prerogative of Hollywood. In the films of the “golden age of Greek cinema”, sexist jokes, the portrayal of women as inferior and weak, racism, homophobia and “body shaming” were key “ingredients” of the successful recipe. Costas Voutsas impersonating the black servant in the 1973 film “The nigger and if you wash him” would be considered the epitome of blackface today and the film a racist delusion. Dimitris Papamichael and Dionysis Papagiannopoulos in “Beating came out of Paradise” would promote violence against women –especially minors. Papamichael has additional mistakes in his potential, as in both films “Beating came out of Paradise” and in “Heartbeat among desks” he enters into a relationship with a minor student.

Figure 3, Costas Voutsas as a "nigger"

The real question is whether it is fair to judge these works in today's social context. Many argue that it is anachronistic. Others believe that since they are still projected on the small screen, we have to point out and correct the wrong texts of the past. What is certain is that with today's data, similar films cannot be shot and released in cinemas. However, in their time, they reflected the manners of the society of that time and the audience was not disturbed. Not only because they were innocent or even naive, but also because the scripts were the mirror of a society that was in no way willing to break it.

2.2.3 Historically Banned and Challenged Plays

In this section, some plays and musicals that have been under criticism by community will be presented or, worse, they have been restricted in American high schools and middle schools (or even colleges in some cases) in the previous years. The majority of the plays and musicals did not have any offensive content and in the rare cases they did, it changed accordingly. Therefore, they were banned or challenged due to the ignorance of the community about the content they promoted.

School administrators of Thespians at Portage (Indiana) High School tried in November of 2015 to steal the references of smoking, drugs, sex and alcohol from a production of their school named as “The Bad Seed”. “The Bad Seed” was a 1954 play from Maxwell Anderson about a child sociopath. As soon as the students discovered what school administrators tried to do, they went to district Board of Education expressing their opinions and beliefs. They said that there was copyright violation and contract breach with the play publisher, if they edited the play without permission. What finally happened was that the production of the play from the students continued as originally planned, as superintendent Richard Weigel said that from his perspective the purpose of the theater is to express how characters act as they have different ways of thinking. Furthermore, Weigel also said that plays, as the aforementioned, constitute an opportunity for students to play these characters but not become these characters.

The drama director at South Williamsport (Pennsylvania) Area Junior/Senior High School in July 2014 said in public that the production of Spamalot has been cancelled due to its content regarding homosexuals. Spamalot is a play of Eric Idle and John du Prez, based on a screenplay by Monty Python. Coming back, the drama director accused the school’s principal for the cancellation. Although the school’s administration said that this was not the real reason of cancellation, internal school emails revealed in August 2014, under Right-to-Know laws, showed clearly that the production was cancelled due to “homosexual themes”. As a result, the drama director was fired four weeks later.

Timberlane Area School District superintendent cancelled a production of Sweeny Todd in Plaistow, New Hampshire in March 2014. Sweeney Todd is a play by Stephen Sondheim and Hugh Wheeler. Students and their parents disagreed with this decision and made a protest having broader support from the community. Finally, they managed to restore the show and reschedule it for production in May 2015.

During the last few years, there have been many challenges and cancellations of Laramie Project. The most recent cancellation occurred in June 2013 in Ottumwa, Iowa. Laramie Project is a play by Moises Kaufman and the Tectonic Theatre Project. Laramie Project was planned for production by Ottumwa High School for June 2013. However, the school's principal did not agree because he believed that the play was too adult-oriented for high school students and he cancelled it. Furthermore, he stated that the play preaches a great message. At the end, the students produced the play in a venue outside their school.

In June 2013, the superintendent cancelled all academic drama programs at the school in Everett of Massachusetts. The reason behind this decision was citing content in student written plays that were presented earlier in 2013. As he said, he found many references to sex and drinking. Furthermore, one actor revealed his underwear after dropping his pants.

Another play that had multiple cancellations was Rent by Jonathan Larson. In November 2013, a planned production for 2014 Spring of Trumbull High School in Trumbull was cancelled by the school's principal. Students and their parents disagreed and protested against this decision. Finally, the musical was restored and the production went on as scheduled. Similar productions of Rent were cancelled in other locations in 2008-2009. Some locations were Newport Beach, California; Bridgeport, West Virginia; Red Wing, Minnesota; suburban Dallas, etc. In California, specifically, the play was cancelled at Corona Del Mar High School because the principal was against the musical's treatment of "prostitution and homosexuality".

Moreover, at Loveland (Ohio) High School in December 2012 a drama director was fired. The reason was that the production of Legally Blonde was too racist, as the school's administration said.

On the other hand, a school production was cancelled during rehearsals due to repeated use of a racist adjective as the play. This case took place at Waterbury (Connecticut) Arts Magnet High School in February 2011, during the production of August Wilson's Joe Turner's Come and Gone. When this case went public, Yale Repertory Theatre pledged to run educational programs at the school in order to integrate the production and its language in the proper context. Students advocated Yale Repertory Theatre's help and the play was finally restored and went in production with a little delay.

In November 2008, Southeast Missouri State University produced a play for Sister Mary Ignatius Explains It All for You by Christopher Durang. A wealthy donor objected keenly in the media about the play because he believed that the play ridiculed and scorned the Christian religion. As a result, the University not only offered a refund to audience members who were offended by the play, but also pledged to review its policies on play selection for season subscriptions. Furthermore, the letter of the complainant to the university's president was published as an op-ed piece in the local paper, which is owned by the family of the wealthy donor. Finally, the school's theatre was named after the complainant, which is a common practice in such situations.

Another school district superintendent cancelled a production of The Tender Yellow Sky by Tim Milhorn at Orland (California) High School in the fall of 2008. Milhorn, who is a veteran teacher that frequently writes plays for school's productions, was six weeks in rehearsals when the school district superintendent cancelled the production. The official excuse from school authorities was the possibility presented to to try committing suicide because the play explored teenage suicide extensively.

In February 2008, a school play was cancelled just three days before the scheduled performance at Sherwood Middle School. The play was Higher Ground about bullying by Portland. According to school authorities, the main topics of the script, that is, bullying, racism, homophobia, and intimidation required a more mature audience than the students. Nevertheless, the play was performed at the Portland Center for the Performing Arts. In the end, both district superintendent and school's principal said to the school board that they had regretted the way they had handled the play and the students.

In a similar case with the last ones, a production was cancelled by the country board of education in Santa Rosa Beach in Florida. The production was Blithe Spirit by Noel Coward, at South Walton High School in Santa Rosa Beach, Florida. The community made a complaint to the country board of education because they believed that the play might "encourage exploration of witchcraft and the occult" and disrupt students' beliefs for monogamous relationships. Eventually, the play was performed in Seaside Repertory Theatre outside the High School campus.

In February 2008, the school's authorities decided to demand changes regarding a play a few hours before scheduled production. The play was Catcalls by Peter Keahey, a teacher at Yellow Springs (Ohio) High School. As a result, all the participants, Keahey, cast and crew refused and, in contrast, they read a letter denouncing censorship denying performing the play. After a short time, the play was produced on the campus of Antioch College.

November 2007, a NAACP complaint cancelled the production of the play “And Then There Were None” by Agatha Christie at Lakota East High School in suburban Cincinnati. The complaint stated that the detective story had a racially insensitive title. The original title of the play was “Ten Little Indians in the United States”. The basis of the play was a novel published in England under a title substituting a racial adjective for “Indians”. The principal decision dropped off two days later and the play was produced as planned.

Wilton (Connecticut) High School scheduled a performance in April 2007 for a play named as Voices in Conflict. This play was changed by the writings of American veterans from Iraq. The play was cancelled by the school’s principal because it did not have a “balanced view of the war”. Finally, in June 2007 two Manhattan theatres invited the school’s drama teacher Bonnie Dickinson to produce the play with his students.

An awkward case took place at John Jay High School in Cross River in New York during March 2007. Three students were suspended after reading in public a selection of the play “The Vagina Monologues” by Eve Ensler. Having Even Ensler personally intervened, the suspension was revoked.

The community made an intervention to the principal of Fulton (Missouri) High School about a production in the spring of 2006. The production was “The Crucible” by Arthur Miller, and “Grease”, by Jim Jacobs and Warren Casey, and the community made a complaint about “immoral behavior” in previous productions. This led the principal to pre-emptively cancel the scheduled production. He stated that he had made that decision in order to avoid additional scrutiny that had already taken place due to the fall of the production of “Grease” (Educational Theatre Association, n.d.).

2.3 Sentiment Analysis

2.3.1 What is Sentiment Analysis?

Sentiment Analysis (SA), which is also called opinion mining or context mining, is extensively applied in Natural Language Processing (NLP), information retrieval, text mining to assist in locating, extracting and classifying subjective information in a systemic way. Sentiment analysis has been an effective tool for processing user comments or reviews especially in electronic commerce or social media. For example, recommendation systems are being used to provide users with views of products or services which match their potential preferences.

Every sentiment notion includes three terms: the object for which someone provides an opinion or context, the attributes or features of this object, and the actual opinion on the object. In reference to these terms, SA then needs to identify the object, extract the features of the object and aggregate opinions on the object. This classification task is conducted on three levels:

a. Document level
b. Paragraph level
c. Feature level

SA at document level assists in defining the overall polarity of a subject regardless of the users' opinions. It assumes that every opinion holder has expressed their view for the subject in this document (e.g. a social media page on a specific subject). SA at paragraph or even sentence level assumes that each sentence contains a different opinion. At this level, SA refers to two specific steps: subjectivity detection and opinion detection. At feature level, any object attributes are analyzed. For example, assuming that a customer buys a car, then they notice that the multimedia system is of high quality but the quality of the interior design is poor. All these data are captured and analyzed separately.

As a process (Figure 4), SA is initiated with data preprocessing (e.g., splitting sentences into words, clearing stop words, stemming), and continues with feature selection and sorting. Then the SA system needs to calculate the polarity of the data (Sharma, Sabharwal, Goyal, & Vij, 2020).

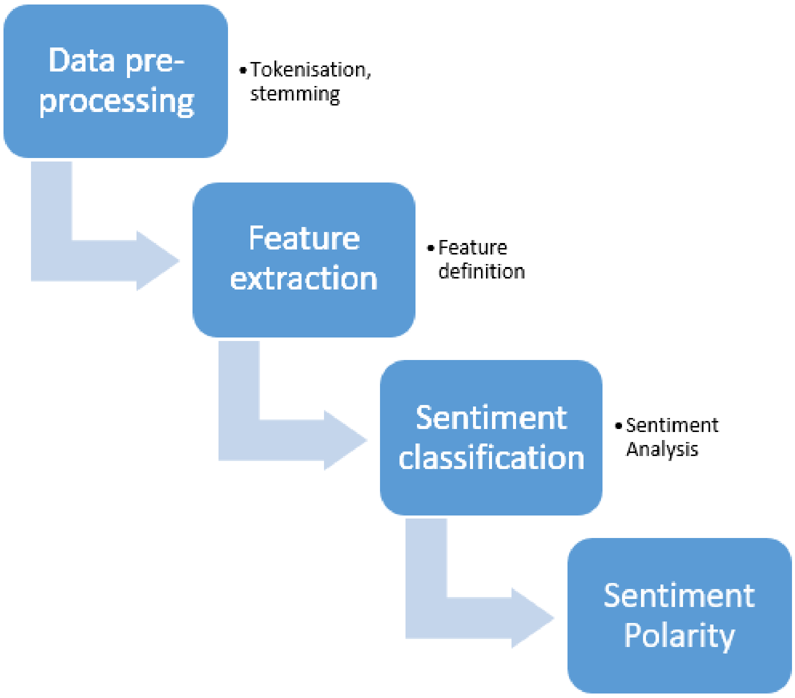

Figure 4, Sentiment Analysis workflow (Sharma, D., Sabharwal, M., Goyal, V., & Vij, M.)

2.3.2 Applications of Sentiment Analysis

The most common applications of sentiment analysis are user reviews of products and services (analysis of brand reputation). (Noble Desktop. (Jan28, 2022) There are many e-commerce sites that provide automated product review summaries (for example Amazon). At the time e-commerce users decide to buy a product, they highly consider the opinion of other consumers which determines the reputation of the product. Sentiment analysis is a valuable tool for revealing users' opinions of certain products, therefore it provides input to e-commerce recommendation systems. SA is able to aggregate opinions around a high-level view of what a group of people think rather than narrow down to a view of a single person which may be misleading. SA can emphasize specific features of the product and aggregate opinions in reference to these features.

Moreover, e-commerce firms should be informed about consumers' views of their products and what to improve to keep them happy. SA can extract those features of the products which are considered to be important for users' decisions (Collomb, Costea, Joyeux, Hasan, & Brunie).

2.3.2.1 Applications to Review-Related Websites

There are websites that ask for user feedback, information and reviews on third party products or services. The objects of review are not limited to product reviews, but they may also include views on services provided by third parties, political issues, etc. There are also applications that aim to attract customers. In this type of applications, the most important problem encountered is the summary of customer views. Additionally, another very important issue is how a user who wants to make a review can amend it in case of errors. Thus, there are important indications that user reviews may be biased or may need corrections. Automated categorizers allow for such updates - corrections (Collomb, Costea, Joyeux, Hasan, & Brunie).

2.3.2.2 Applications as a Sub-Component Technology

The systems of SA and opinion mining play an important role as they provide the necessary technology for some other systems. Thus, SA systems may aim at strengthening referral systems. That is, a system to avoid suggesting products or services for which it has received negative information. Another way of subjectivity detection or categorization is the identification of abusive language in e-mails or other types of communication .

In addition, in online systems that contain advertisements as sidebars, it is very helpful to find web pages that contain sentimental content which is not suitable for incorporation within advertisements. For more sophisticated systems, it would be useful to have product advertisements, if and when a positive sentiment has been detected, and it may be even more important to exclude ads of products for which a negative trend has been detected.

It is now accepted that information mining can be improved with the use of subjective propositions, namely by detecting negative information within these proposals. Providing answers to questions is another area in which SA may prove useful. Researchers suggest that questions, when provided with answers which mainly include information about how an entity is judged by the general public, can provide users with a lot of useful input.

In addition, the references analysis is important, where for example, one may be interested in why an author suggests a piece of work which they reject. Similarly, an attempt is made to use the semantic orientation to extract value according to the reviewer. In conclusion, the computational processing of sentiments is partly motivated by the objective to improve human-computer interaction.

In summary, SA is a valuable tool for recommendation systems (for example, suggesting products or adverts based on preferences or opinions) as well as other capabilities. For example, it helps to improve data mining by eliminating subjective sections of documents. In addition, it guides the adaptation of customizable User Interfaces to users' preferences or opinions removing non-relevant elements (Collomb, Costea, Joyeux, Hasan, & Brunie).

2.3.2.3 Applications in Business and Government Intelligence

The areas of opinion mining and sentiment analysis are compatible with different types of intelligence applications. Indeed, business intelligence seems to be one of the main factors behind the cooperative interest in this area.

Considering a simple scenario which involves a large computer manufacturer, who is frustrated by the unexpectedly low sales and is faced with the question: "Why do customers not buy the company's laptops?". Specific factors such as the weight of the laptop or the price of a competitor's model are entirely relevant. To answer this question, the specific company needs to focus more on people's personal views on these features. In addition, subjective views regarding intangible properties -e.g., "design is beautiful" or "customer service is good"- or even misconceptions -e.g., "updated device instruction programs are not available"- should also be considered.

Sentiment analysis technologies for extracting opinions from unstructured documents are excellent tools for handling many business operations. In reference to the above scenario, it would be difficult to investigate directly the reasons why customers do not buy the manufacturer's laptops. Instead, they could use one system which (Collomb, Costea, Joyeux, Hasan, & Brunie):

1. would collect reviews or other similar opinions from discussion groups on the Internet, blogs and sites and
2. would then group the reviews and individual ratings into a summary review to report users' overall feeling.

This would help analysts to avoid going through possibly hundreds of reviews (e.g., complaints). In addition to a summary review of the product reputation, the public's view of the company could target towards making forecast plans related to the development of sales or marketing in reference to the collected data.

2.3.2.4 Applications Across Different Domains

As is well known, opinions play a particularly important political role. Some research has focused on understanding what voters think, while other projects have as a long-term goal the clarification of politicians' positions, i.e., what views they support, such as public supporting elements or rejecting in order to enhance the quality of the information accessed by voters.

Sentiment analysis has been proposed as key to applications (e.g., Twitter) on which people submit their views about upcoming policy or government proposals, allowing automatic analysis of their views.

The interaction with sociology is extremely helpful. For example, the question of how ideas and innovations are disseminated includes the question of who is positively or negatively disposed towards whom, and therefore who will be more or less receptive to new information which will stem from a specific source. To cite another example: structural equilibrium theory basically deals with the polarity of "bonds" among people and how they affect the cohesion of a group. These ideas have started being implemented through analysis of online media (Collomb, Costea, Joyeux, Hasan, & Brunie).

2.3.3 Sentiment Analysis Research

The aforementioned real-life applications of sentiment analysis are only part of the reason why SA has become very popular in the research field. Nowadays, it is considered to be a challenging topic in the area of Natural Language Processing research. SA has become more popular after the year 2000. This was due to the fact that there was no sufficient textual opinion in digital format. After the rapid development of social media, blogs and news sites, it has become a key in research areas related to NLP, data mining, and information retrieval deepening into three levels (Liu, 2012).

2.3.3.1 Sentiment Analysis Levels

2.3.3.1.1 Document level

At this level, an SA tool aims to classify an entire document which contains textual opinions of whether it expresses a positive or negative sentiment. For example, this may be related to a product review. The system or tool assumes that each document contains opinions for a single item (Liu, 2012).

2.3.3.1.2 Sentence level

At this level, the goal of the specific tool is to determine whether each sentence contains a positive or negative view (or sometimes a neutral one). The classification effort at this level focuses on identifying subjectivity and making a differentiation from sentences which provide actual information. Subjectivity is not supposed to point towards sentiments since many objective propositions can imply certain opinions, e.g., “I bought the phone last month and the power button is not working” (Liu, 2012).

2.3.3.1.3 Feature level

However, the document and sentence level do not provide in depth information about people’s sentiments. The feature level is targeted towards the analysis of language parts in order to examine opinions for certain attributes of a product or any other entity. This may be hidden within documents, paragraphs, sentences or phrases. Analysis at this level is structured in two terms: sentiment emotion (positive or negative) and goal (opinion). In other words, every opinion is bound to a purpose in order to mark its importance. For example, although the sentence “although the floor is not so good, I still want to buy this house” has a clearly positive connotation, we cannot determine a purely positive opinion. The sentence is positive regarding the house but negative concerning the floor.

In many applications, opinion is targeted towards specific features of products or services. Thus, at feature level the analysis effort is on discovering emotions regarding the aforementioned items. For example, the phrase "the garden space of the house is good, but the floor is of low-quality wood” evaluates two features, garden and floor. The emotion concerning the garden is positive, whereas the one about the floor is negative. The quality of the garden and floor is the target of the opinion. As a result of this analysis, a structured summary of views on features is generated which provides both a qualitative and quantitative report (Liu, 2012).

2.3.3.2 Sentiment Lexicon

As mentioned above, the most important indicators of sentiments are phrases or words which express positive or negative sentiments. For example, good, wonderful, poor, rich. A list of such words and phrases is called a lexicon of sentiments or opinions. The generation of such lexicon is a challenging research topic for multiple reasons (Liu, 2012):

1. A word indicating positive or negative emotion can have opposite orientations in different application areas. For example, “sucking” usually indicates negative emotion, e.g., “This phone sucks”, but it may also imply positive emotion, e.g., “This vacuum cleaner really sucks”.
2. A sentence that contains words of emotion may not express any emotion in particular. This often occurs in different types of sentences. For example, “Can you tell me which phone is good?” and "If I find a good phone on Black Friday, I will buy it.”. Both sentences contain the word “good”, but neither expresses positive or negative feedback for any particular phone. However, not all conditional sentences or questions express feelings.
3. Sarcastic sentences with or without emotion words are difficult to deal with, e.g., “What a wonderful phone! Its battery does not last more than a day”. Sarcasm is not so common when it comes to consumer reviews of products and services, but it is quite frequent in political debates, which makes political views more difficult to analyze.
4. Many sentences without words of emotion can also imply certain opinions. Many of these are in fact objective sentences which are used to express some factual information, however, there are still many types of these sentences. For example, “this washing machine uses a lot of water” implies a negative feeling about the washing machine because it consumes a lot of water. The sentence “After you sleep on a mattress for two days, a hollow has formed in the middle” expresses a negative opinion about the mattress. This phrase is objective and indicates a fact. All these sentences contain no words of emotion.

2.3.3.3 Natural Language Processing Issues

As an NLP problem, sentiment analysis is challenging many sub-issues of NLP, e.g., clarification of the concept of the word, which remains an unsolved NLP problem. For any such SA tool there is difficulty in fully understanding the semantics of each sentence in a document. Certain aspects of its sentences or phrases may be comprehensible and positive or negative emotions on specific subjects might be detected. NLP researchers have been substantially working on this area (Liu, 2012).

2.3.4 Sentiment Analysis Models

Inspired by the shaping of the human brain, sentiment analysis algorithms mimic the way the human brain processes data through an artificial neural network. Powerful neural network algorithms are used by NLP Deep Learning techniques in order to perform sentiment analysis. It is an emerging technology used to detect whether a piece of text is positive, negative or neutral. It carries out this process by assigning sentiment scores to categories, topics or entities. These categories could be specific stores, products, pricing strategy, locations, offers, etc..

As already mentioned, sentiment analysis is widely used to evaluate customers' views on the internet. A new case involves companies that have to deal with the consequences of high churn rates. HR teams use data analytics combined with sentiment analysis to understand what employees say to reduce turnover and improve performance. This is not as personalized as locating a specific person, but it is used in order to understand general trends and assume corrective action if needed. Up until now, the importance of this technology has been proved. If anyone considers integrating it into their company analytics system, it is beneficial to understand what is required.

Sentiment lexicons are made up of many dictionaries that have an exhaustive list of phrases and adjectives which have been manually graded in advance. This is how we understand phrases. The first time we hear a phrase, we may not understand it, but based on the context in which it is used, we categorize it in our brain whether it has a positive, negative or neutral connotation.

Sentiment lexicons follow the same pattern, but human coders manually rate each of these phrases. This can be quite difficult because everyone has to agree on the rating to be given. For example, if one person gives an "awful" score of 0.5 and another person gives a "dislike" score for the same score, then the sentiment analysis will find that both words have the same negative intensity. However, it is quite obvious that "awful" should go beyond "dislike".

If a multilingual sentiment machine is required, then unique lexicons for each language are necessary. Each of these lexicons must be maintained, scores are also modified and new phrases are added or removed. Nowadays, there are ready-made lexicons that only need to be improved. Once someone selects and creates the lexicon of their preference, they need to define the algorithm they will use for sentiment analysis. They will also be able to choose one or more algorithms belonging to three major sentiment analysis algorithm models. Finally, they can select any model they might want, depending on the amount of data to be processed and the accuracy required (iTech, 2021).

2.3.4.1 Rule or Lexicon Based Approach

This approach is based on manually created rules for classifying data in order to determine sentiment. It uses a list of words with positive or negative values to indicate the polarity and power of sentiment in calculating sentiment score. Additional functionality can also be added including expressions. Rule-based sentiment analysis algorithms can be adapted to the environment by developing even smarter rules. Rule or lexicon-based approach calculates sentiment score in a two-step process:

1. At first, the number of positive and negative words in the given text is counted.
2. Secondly, sentiment score is calculated by using an algorithm, like the following ones:
    a. If the number of positive words is greater than the number of negative ones, a positive sentiment will be generated.
    b. If both are equal, a neutral sentiment will be generated.

Despite the fact that it is a quite straightforward approach, it has some disadvantages. Its major disadvantage is that it does not take into account how words are combined in a sentence; it only evaluates facts. It is an approach that is implemented quickly, but the model entails long-term cost, as it requires regular maintenance in order to produce consistent and improved results (iTech, 2021).

2.3.4.2 *Automated or Machine Learning Approach*

This approach uses machine learning to understand the essence of the statement. This ensures that the accuracy of the analysis is improved and the information can be processed according to many criteria without being too complicated. The specific approach involves the use of supervised machine learning algorithms. An algorithm is trained with many sample snippets until it can accurately predict the emotion of the text. Large chunks of text are then fed into the classifier and it is able to predict the sentiment as negative, neutral or positive. Machine learning models can be of two types (iTech, 2021):

- Traditional models: these methods require the collection of a data set with examples for positive, negative and neutral classes, then process these data and, finally, train the algorithm based on the examples. These methods are mainly used to determine the polarity of the text. Traditional machine learning methods such as Naïve Bayes, Term Frequency-Inverse Document Frequency (TF-IDF), Logistic Regression and Support Vector Machines (SVMs) are widely used to analyze large-scale sentiment because they are capable of scalability.
- Deep Learning Models: these models provide more accurate results than traditional models and include neural network models such as CNN (Convolutional Neural Network), RNN (Recurrent Neural Network) and DNN (Deep Neural Network). (Nhan Cach Dang, María N. Moreno-García, Fernando De la Prieta, 2020)

2.3.4.2.1 Traditional Models

In this section, two of the most common traditional models for sentiment analysis are going to be described. These models are going to be implemented for the purposes of this analysis.

2.3.4.2.1.1 Naïve Bayes

This model applies Bayes' theorem with a naive assumption that there is no relationship between different features. According to the Bayes' theorem (Sharma M. , 2020):

$$Posterior = likelihood * proposition/evidence$$

or

$$P(A|B) = P(B|A) * P\ (A)/P\ (B)$$

An example will be presented in order to facilitate understanding of this model. Let's assume that in a deck of playing cards, a card is chosen. Therefore, what needs to be found is the probability of a card being the queen given the fact that the card is a face card. This example can be solved using Naïve Bayes' theorem as follows:

$$P(QueengivenFaceCard) = P(Queen|Face)$$

$$P(FacegivenQueen) = P(Face|Queen) = 1$$

$$P(Queen) = \frac{4}{52} = \frac{1}{13} P(Face) = \frac{3}{13}$$

Having an input with several variables like these:

$$P(y|x_1, x_2, \dots, x_n) = P(x_1, x_2, \dots, x_n|y) * P\ (y)/P\ (x_1, x_2, \dots, x_n)$$

Using Naïve Bayes, we can assume that $x_1, x_2, \dots, x_n$ are independent from one another, e.g.:

$$P(x_1, x_2, \dots, x_n|y) = P(x_1|y) * P(x_2|y) * \dots * P(x_n|y)$$

The assumption in the distribution of $P(x_i|y)$ leads to different Naïve Bayes' Models (NBM). For example, assuming that the Gaussian distribution will lead to Gaussian Naive Bayes (GNB) or that the polynomial distribution will yield Multinomial Naive Bayes (MNB).

Naive Bayes' Model works particularly well with text sorting and spam filtering. The advantages of working with the NBM algorithm are (Sharma M. , 2020):

- It requires a small amount of training data to learn the parameters.
- It can be trained relatively quickly compared to advanced models.

The main disadvantages of the NBM algorithm are:

- It is a decent classifier but a bad evaluator.
- It works well with discrete values, but it does not work with continuous values (cannot be used in regression).

2.3.4.2.1.2 Term Frequency-Inverse Document Frequency

As it is widely known to the data science world, statistical approaches like machine learning and deep learning perform well with numerical data. Nevertheless, in the real world natural languages consist of words and sentences. Therefore, the very first thing any model needs to do, prior to creating a sentiment analysis model, is to convert the text into numbers. For this reason, many different approaches have been implemented, such as Bag of Words, N-grams, Word2Vec and TF-IDF.

The Term Frequency-Inverse Document Frequency (TF-IDF) algorithm is used to evaluate the importance of words in a given text. The importance is proportional to the number of times the words appear in the given text and inversely proportional to the frequency of words appearing in the given text. At this point, it is worth mentioning that in a simple Bag of Words, each simple word would have the same importance compared to the other ones. The idea behind TF-IDF is that words that appear more often in a text and less often in other texts should be more relevant because they are more useful for classification.

TF shows the frequency of words, e.g., the number of times that appear in a given text, and it is calculated as follows:

$$tf_i = \frac{n_i}{\sum_k n_k}$$

The previous function calculates the number of the word occurrences out of the total number of words existing in the document.

IDF is a metric system of the importance of the word in the whole document. As it can be seen in the following function, it is the logarithm of the inverse of the proportion of documents in a given text containing the word. This contains the total number of documents in the given text over the number of documents in which the word is present. The value of IDF is the logarithm of this result:

$$idf_i = \log \frac{D}{|\{d_j : t_i \in d_j\}|}$$

As a result, the value of TF-IDF weight is calculated by multiplying values found for TF and IDF. The higher value in TF-IDF weight for a word under examination means greater importance of this word within the given text (Chiny , Chihab, Chihab, & Bencharef, 2021).

$$tfidf_i = tf_i * idf_i$$

2.3.4.2.2 Deep Learning Models

In this section, two of the most common deep learning models for sentiment analysis will be described.

2.3.4.2.2.1 Convolutional Neural Networks

One of the most known and widely adopted models regarding in-depth learning in order to classify texts is the Convolutional Neural Network (CNN). The reason that CNN is so popular is because it takes advantage of the so-called convolutional filters that automatically learn features suitable for the task at hand. For example, if CNN is used to classify sentiment, convolutional filters can capture intrinsic syntactic and semantic features of sentimental expressions. It has been shown in many studies that a single convolutional layer, a combination of convolutional filters, can achieve comparable performance even without any special hyperparameter tuning. In addition, CNN does not require any special knowledge of the linguistic structure of a target language. For these reasons, CNN has been successfully inserted into various text analyses: semantic analysis, query search, sentence modeling, etc..

One could argue that the Recurrent Neural Network (RNN) may be better for text classification than CNN, as it maintains the order of the word sequence. However, CNN is also capable of recording sequential patterns, in terms of local patterns from convolutional filters. For instance, convolutional filters along with the attention technique have been successfully applied to machine translation. In addition, compared to RNN, CNN has mainly a smaller number of parameters, and, as a consequence, CNN is trainable with a small amount of data. CNN is also known to explore the richness of pre-trained word incorporation (Kim & Jeong, 2019).

2.3.4.2.2.2 Recurrent Neural Networks

Recurrent neural networks (RNNs) are artificial neural networks modeling dynamic system behaviors by using hidden states. They have been the answer to most consecutive data and natural language processing (NLP) problems for many years. This was due to the fact that traditional neural networks receive a constant amount of input data each time and produce a constant amount of output every time. In contrast, RNN do not consume all inputs at once. Instead, they take them one by one, sequentially. At each single step, RNN perform a series of calculations before generating an output. Then the output, which is called hidden mode, is combined with the next input in the sequence in order to produce another output. This process continues until the model is scheduled to terminate or end the input sequence. However, a major drawback affecting the typical RNN is the problem of slope disappearance/explosion. This problem occurs during backward propagation via RNN during formation, especially for deeper layer networks. For this reason, LSTM was proposed in 1997 (Chiny , Chihab, Chihab, & Bencharef, 2021).

2.3.4.3 Hybrid Approaches

Hybrid sentiment analysis models are the most modern, effective and widely used approach to sentiment analysis. As long as you have well-designed hybrid systems, you can really gain the benefits of both automated and rule-based systems. Hybrid models can offer the power of machine learning combined with the flexibility of customization (iTech, 2021).

2.4 Machine Learning

Machine Learning techniques are used to classify a sentence into a category. The techniques emerged and developed through the study of Pattern Recognition and they are constantly evolving. Both fields are subfields of Artificial Intelligence (AI) in Computer Science. Throughout the years, plenty of definitions have been given for Machine Learning. First and foremost, it is imperative to study the way the human brain learns. It tries to understand its environment; it observes it and creates a simplified (abstract) version of it called a model. In addition, it has the ability to organize and relate its experiences and performances by creating new structures called patterns. By transferring these concepts to the field of computer science, it could be defined that "Creating models or templates from a set of data, within a computer system, is called machine learning".

Machine learning techniques and used algorithms are classified into four main categories and are used according to the nature of the problem (Neethu & Rajasree, 2013):

1. Supervised Learning

2. Semi-supervised Learning
3. Unsupervised Learning
4. Reinforcement Learning

2.4.1 Supervised Learning

Equally referred to as learning with supervision, this technique and the training of the algorithms require the existence of a data set with ordered polarity (Training Set). Using the training set as a basis (model), supervised learning takes a large amount of data, analyzes it, draws standards and conclusions and finally leads to predictions. An input data set is created (Training Set) in a way that the output is known. That is, for each of the input data, the emotional orientation is known (classified polarity, category, class or label). The algorithm is based on the Training Set and the following steps take place:

1. algorithm is fed with input data of unknown polarity, which are texts in natural language,
2. it processes them,
3. it tries to extract templates that match the set of training,
4. it categorizes / characterizes the input data and
5. finally draws conclusions with characteristic polarity that can lead to predictions.

Supervised learning is used in problems of classification, regression and interpretation. Some indicative examples of supervised learning algorithms are the Naive Bayes and Support Vector Machine (SVM) algorithms (Ahmad, Munir & Aftab, Shabib & Muhammad, Syed & Awan, Sarfraz., 2017).

2.4.2 Semi-supervised Learning

In this category, the Data Set may be labeled or unlabeled. The algorithm will be trained in order to make decisions based on the information given. (Ahmad, Munir & Aftab, Shabib & Muhammad, Syed & Awan, Sarfraz., 2017)

2.4.3 Unsupervised Learning

Otherwise referred to as non-supervised or unattended learning, the algorithm in this case does not provide any information and has to discover the structure, patterns or correlations of the data itself. It is used in association analysis and clustering problems. The most well-known non-supervised learning and classification algorithms are clustering algorithms (k-Means, k-Nearest Neighbors, etc.) (Ahmad, Munir & Aftab, Shabib & Muhammad, Syed & Awan, Sarfraz., 2017).

2.4.4 Reinforcement Learning

It is used in planning problems, such as robot motion control, autonomous vehicle movement, factory optimization and learning board games - mainly strategic ones, such as chess or Go28.In reinforcement learning (RL) tasks, the agent conceives the state of the environment, and acts in order to maximize the long-term return, based on a real valued reward signal. RF algorithms have been applied to many text-based games for the purpose of extracting the mindset of a player or to classify the sentiment at the time each move occurs. (Garcıa, J., & Fernández, F., 2015)

2.5 Sentiment Analysis on Movie Scripts and Plays

2.5.1 Background

Sentiment analysis, as a field of study, has seen a significant increase in its application in various fields, as it can easily become interdisciplinary and facilitate different tasks. It has been established as a respected factor of text analysis due to its simplicity and plain approach. Some areas in which sentiment analysis has been successfully implemented are business systems, marketing campaigns and recommender systems.

For some years now, sentiment analysis combined with NLP and ML techniques has been used in a number of different applications for movie scripts and their reviews. They have been used to locate patterns in the structures of the movies. Furthermore, they have been used to learn to predict the upcoming emotional state based on the previous one, respectively showing that “successful” movies follow specific narrative developments and have a specific “flow” and consistency in the way emotional states unfold. In addition, it has been observed that existing techniques for analyzing binary (“positive /

negative”) emotions have a “positive bias”, favoring the learning of positive emotions over negative ones, thus “underestimating” the existence of the latter. This observation can create a significant deviation in accuracy of about 10 % to 30 %. It turns out that taking into account the meta-characteristics (capital letters, punctuation marks and parts of speech) helps to resolve this problem.

Some other approaches have attempted to classify reviews based on words that were semantically similar. This has been done in order to identify communities of reviewers via clustering. Other alternatives that had interesting points suggest identifying the driving aspects of a movie which are mainly related to the script, the interpretation and the plot or some more specific features (e.g., music, effects) and how they are reflected in the reviews of sentiments. Results show that the most important factors influencing a review are usually acting and plot. Nevertheless, variations in these studies that examine the same aspects in different types of movies may change the factors that lead to the corresponding reviews. The most typical methodology that is used to perform sentiment analysis directly involves classic text mining solutions, such as extracting N-Gram or adding Part-of-Speech tags in conjunction with Naïve Bayes (NB) or Recurrent Neural Networks (RNN). However, different techniques, like GINI Index usage or Support Vector Machine (SVM) approaches have increased the accuracy of the results. The Skip Gram and Continuous Bag of Words (CBOW) models also have promising results.

A recent research project has tested a number of techniques to extract emotion from reviews. In this project, a combination of emotion lexicon and word incorporation were used to capture the critics' emotion. Using this combination, a satisfactory level of predictability for the critics’ film binary rating was obtained which, in addition, provides solutions for both English and Greek datasets. Regarding the classification of movie scripts based on NLP, attempts have been made to use subtitle files of different movies to finally predict their genre. The premise is that, through sentiment analysis in the subtitle environment, sentiment can be extracted and this can lead to the determination of the type of film. Preliminary results show that these techniques work best when applied to action, romance or horror films. As mentioned before, these are preliminary results and further analysis is required in order to strengthen the forecast.

The emerging importance of sentiment analysis in film and critique evaluation has also prompted well-known and respected online competition networks, such as Kaggle, to take initiatives in order to propose innovative solutions to SA problems. It is obvious, however, that in order to produce a well-documented solution, the characteristics must be carefully chosen, taking into account the variation of the terms, the handling of the negation and the treatment of the words related to opinion (Frangidis, Georgiou, & Papadopoulos, 2020).

2.5.2 Related Work

What is necessary in the analysis of the narrative is emotion. It is necessary because, for human beings, there is a wide range of different emotions when reading or experiencing a narrative. Following in this section, previous work related to emotions analysis, especially in storytelling, will be presented (Hye-Yeon, Moon-Hyun, & Bae, 2017).

2.5.2.1 Emotion Analysis for Narrative

When someone reads a text story, their emotional reactions may be divided into two types:

- Cognitive interest: this type of interest represents the emotions that can appear as cognitive reactions mainly due to the narrative structure, which includes emotions such as suspense, surprise and curiosity.
- Emotional interest: this type of interest expresses the emotions that a reader may feel when entering a world of history, including emotions such as character identification or empathy.

Although the previously mentioned types of literary emotions and the reader's emotional reactions to narrative are often intertwined and difficult to distinguish, they can be a convenient starting point for dealing with narrative-related emotions.

Despite the fact that emotion is a complex human phenomenon involving mental, psychological, physiological and biological activities, the majority of emotion models can be divided into two types:

- Distinct or basic emotion models: in this type of models, many fundamental emotions exist (e.g., Ekman’s six basic emotions –happiness, sadness, anger, fear, disgust and surprise) that are primitive in nature (regardless of age, gender or race).

- Two-dimensional or three-dimensional models: in a two-dimensional (or circumflex) model there can be presented a plethora of different emotions through two dimensions (e.g., vigor and arousal). On the other hand, in a three-dimensional model there can be presented a variety of emotions through three dimensions (e.g., pleasure [or vigor], excitement and dominance).

Moreover, the OCC emotion model (Ortony, Clore, and Collins) is often used in computational emotion models. Twenty-two types of emotions are described as tacit reactions to the cognitive assessment of a factor for a given situation in the OCC emotion model. While the OCC emotion model can be categorized as a kind of distinct emotion model, it does not differentiate between basic or fundamental emotions and compound or complex emotions (Hye-Yeon, Moon-Hyun, & Bae, 2017).

2.5.3 Movie Script Structure

As it is already known, movie scripts are mainly technical documents created by screenwriters. They are the main reference for the director of a movie and they consist of scenes that represent key story sections in a coherent plot. Whenever there is a change in location or time, a new scene appears. Furthermore, some conventions exist in movie scripts in order to describe scenes more precisely. Some of the conventions are the headings at the beginning of each scene that determine if the location of the scene is internal or external and the specific location where it takes place (e.g., supervisor's office or City Hall). Moreover, the time of the day can be determined at the beginning of each scene (e.g. afternoon).

Based on experiential data, two types of text can be contained in movie scripts: text that provides a general background to the scene (see example 1) and dialog that can be rendered directly to one of the characters (see example 2). Each dialog may contain additional information provided in parentheses (see Example 3). In addition, the movie script may contain some technical information, such as CLOSED SHOT or CAMERA MOVEMENT explaining the movie production process (Anikina, 2017).

(1) WESTLAKE PARK (MCARTHUR PARK) Duffy is paddling, Gittes is sitting on the stern. "They pass Mulwray and a slender blonde girl in a summer dress

with a print, dragging them on their rowing boat, Mulwray was tending to the girl."

(2) J.J. Gates: "Tell me, how many people do you exclude per week?" UNK_other_customer "We do not publish a newspaper entry, I can tell you that."

(3) J.J. Gittes: "(to Duffy, as they pass) Let's have a big smile, man."

3 Case Study – Implementation

3.1 Overview

The case study that is going to be implemented aims to discover racist and homophobic content in scripts from either movies or theatrical plays. In order to implement such a case study, some initial steps are required. First of all, it is essential to decide which programming language is going to be used. Secondly, it is highly important that the source of script be determined. Last but not least, a methodology that will process scripts and decide if they contain racist or homophobic content based on some predefined criteria needs to be created. At this point, it is worth mentioning that these predefined criteria will come up after a number of trials.

The programming language to be used is Python. Python is a free and open source language, having a vast library support. It is mainly used for analyzing and processing data as well as drafting reports on the produced findings. Therefore, in this case, it is an ideal programming language as results from a big amount of data are going to be parsed, analyzed and presented. Furthermore, the existence of numerous libraries, which are extremely handy for implementing a sentiment analysis model, is one additional benefit of the specific choice. Some of the most well-known Python libraries for sentiment analysis are:

- NLTK (Natural Language Toolkit)
- SpaCy
- TextBlob
- Gensim

In this case study, Python and sentiment analysis techniques will be used. A lexicon-based approach is going to be implemented in order to discover the count and the percentage of lines in a script containing positive, negative or neutral sentiments. Furthermore, results that will come up using both figures (mainly plots) and tables (in an Excel file workbook) will be demonstrated. Finally, the term frequency - inverse document frequency algorithm (TF-IDF) will be used in order to train and evaluate this model. The implementation will be based on a hybrid sentiment analysis model, because we use a lexicon-based approach and a traditional machine learning model.

One of the main tasks that are going to be implemented is the calculation of the sentiment score in order to be able to classify a script as either positive, negative or neutral. This task will be done using the lexicon-based approach. Sentiment score can be calculated either by using packages offered by Python libraries or by implementing the specific logic through code. In this case, sentiment analysis will be conducted using four different sentiment scores. The very first three sentiment scores will be calculated using our own custom logic and the last one will be calculated using TextBlob library.

The reasons for creating the particular sentiment score methods are:

1. They are custom-made, through designing and implementing the code that calculates the sentiment score.
2. It is flexible to change the logic based on specific needs, e.g., positive and negative lists of words focused to detect specifically racist and homophobic content can be used.
3. We can have multiple features for analysis when we use multiple approaches to calculate the sentiment score.

As a fourth choice for sentiment score calculation, TextBlob was selected because it is very user-friendly and has a less oppressive learning curve. Furthermore, it is an attractive and lightweight Python library for NLP and sentiment analysis development.

After having calculated sentiment scores using the lexicon-based approach, the implementation will proceed by applying a machine learning approach. The Multinomial Naïve Bayes algorithm in conjuction with the TF-IDF score will be used, since it is one of the most known and widely adopted approaches for machine learning.

3.2 Methodology

The methodology that was implemented is mainly based on the two different approaches which were used for sentiment score calculation (custom implementation and library usage). The common denominator in either approach are the movie and play scripts that are going to be examined. Scripts, especially for movies, can be found online and they consist of several thousands of lines. One of the most known websites is “The Internet

Movie Script Database (IMSDb)[1]". In this implementation, the code aims to find scripts in text format in predefined locations. Scripts should be available in text format in order for Python code to parse them and perform sentiment analysis.

One thing that needs to be clarified is that this implementation, using four different ways for sentiment score calculations, will provide four different results for each movie, showing the percentage of neutral, positive and negative lines among the count of the relative words. The implementation that is going to be used will create a table based on Table 1, containing the results of sentiment score calculations for any movie in one place. Using the results of the aforementioned table, anyone can see the percentages of sentiments and make comparisons between different movies and plays. Apart from tables containing results, the specific code produces figures with plots visualizing the results.

	SC_1 Count	SC_2 Count	SC_3 Count	SC_4 Count	SC_1 %	SC_2 %	SC_3 %	SC_4 %
negative								
neutral								
positive								

Table 1, Sample results for a movie

After applying each of the four sentiment score methods, the TF-IDF approach will be applied in order to evaluate and train the model which consists of the results created by the four sentiment score methods that were used. The aforementioned approach will be applied separately to the results of each sentiment score method in order to identify which method is best suited for every particular case.

With a view to processing scripts and conducting sentiment analysis tasks, the following methodology was followed:

- Configuration read: When the application starts, it reads a configuration file existing in a predefined location. The values read from the configuration file are:
    a. Stop words: These are words in a comma-separated format occurring frequently, but not containing any sentiment value, e.g., "the, to", etc.

[1] URL: https://imsdb.com/

b. Positive words: These are words in a comma-separated format used only in our custom calculation of sentiment score. These words carry positive sentiment value.
c. Negative Words: These are words in a comma-separated format used only in our custom calculation of sentiment score. These words carry negative sentiment value.
d. Movie list: It is a comma-separated list containing the movies that our implementation will process in order to conduct sentiment analysis. For every single movie a section (as it is named in Python) should exist in the same configuration file and it will contain the following values:
    i. Movie name: It is the name of the movie that will be used in the various output forms of the application, such as figures, Excel sheets, etc.
    ii. Script path: This is the path where the script exists in text format.
e. Visualizations Export: It is the name of the Excel file where the application will save the result of sentiment analysis.
f. TF-IDF parameters: These are the values used in the Multinomial Naïve Bayes - TF-IDF approach in order to train and evaluate the model. Apart from the values used, one variable is present, showing the name of the file that will contain all the produced metrics.

- For every single movie found in the previously read configuration file, the following actions will take place:
    a. Movie script load: The very first action that should take place is the loading of the movie script. This is essential in order to load the whole script that is going to be processed onto the application memory and perform sentiment analysis. Although this task may be heavy and memory intensive, as a script may contain several thousands of lines, it is essential for the actions that will take place in the following steps.
    b. Data pre-processing: When the script is loaded onto the application, some preprocessing tasks need to be performed. These tasks are essential as the original script may contain punctuation, stop words, numbers, etc. that are not useful in determining the sentiment. The tasks that will be executed are:
        i. Word conversion to lower case

ii. Punctuation removal
iii. Space removal between words
iv. Number removal
v. URL removal
vi. Stop words removal
vii. Tokenization: In this task, text file lines will be split into words. Tokenization is a mandatory process within the methods to be implemented for calculating sentiment score.

c. Word frequency distribution (optional): It will be useful in some cases to calculate word frequency distribution. In this case, it is useful in the subsequent task, where it is necessary to present the word cloud.
d. Word Cloud presentation (optional): In this task, a figure of all words based on their own frequency in the plot will be presented. The most frequent words will be depicted in a more intensive way, such as in bold, bigger font size, etc.
e. Sentiment Analysis: Sentiment analysis will be conducted using four different methods for sentiment score calculation. For every method, different results will be produced in order to have a plethora of results that will help determine the best fitting sentiment score method for the specific case.
f. Visualizations: One of the most important actions in every case study or project is the presentation of results. The most preferred ways for presentation purposes are figures and tables, because they visualize results in a way that is better understood by everyone.
g. TF-IDF: Finally, the TF-IDF algorithm will be applied to the results of each sentiment score method that was used. Metrics will contribute to understanding which sentiment score method is the most suitable for the particular case. The metrics produced through this implementation are:

i. Precision: it is also known as positive predictive value and it is the fraction of relevant instances among the retrieved instances.
ii. Accuracy: it is the fraction of decisions (relevant / irrelevant) that are correct.

iii. Recall: it is also called as sensitivity and it is the fraction of the total amount of relevant instances that were actually retrieved.
iv. F1 Score: it is also known as F-score or F-measure and it is a measure of a test's accuracy. It can also be described as the harmonic mean or weighted average of precision and recall.

Apart from metrics that will be presented in tables, this implementation will produce a classification report and a confusion matrix for each sentiment score method used in every single movie.

3.3 Sentiment Scores

Four different methods for sentiment score calculation will be used. The first three methods are custom-implemented and the fourth one uses TextBlob library. As far as custom-implemented methods are concerned, it is initially necessary to calculate the number of positive and negative words existing in the script. For this reason, each separate line (Tokenization task) needs to be tokenized and these tokenized words will be compared to user-defined lists of positive and negative words. These user-defined lists exist in the configuration file and are different for positive and negative words. Upon the finalization of the comparison, the amount of positive and negative words in each line will be demonstrated.

For methods one, two and four the following pseudo-code for sentiment score calculation was used:

- $if (SentimentScore) isgreaterthan 0 thenSentimentispositive$
- $if (SentimentScore) islowerthan 0 thenSentimentisnegative$
- $if (SentimentScore) isequal 0 thenSentimentisneutral$

Whereas the pseudo-code for sentiment score calculation for the third method is modified as follows:

- $if (SentimentScore) isgreaterthan 1.5 thenSentimentispositive$
- $if (SentimentScore) islowerthan 0.5 thenSentimentisnegative$
- $if (SentimentScore) isbetween 0.5 \wedge 1.5 thenSentimentisneutral$

Sentiment score is calculated for every single line of the script. Having calculated the sentiment score for every single line, occurrences of different sentiments are calculated.

3.3.1 Method 1: Absolute sentiment score calculation by using positive – negative count

In this method, the sentiment score is calculated using the following type:

$$SentimentScore = \sum(positiveword) - \sum(negativewords)$$

As it is obvious, this method is direct and simple regarding sentiment score calculation. Although it is simple to understand and a straightforward method, it may favor longer lines as they tend to have more counts of positive/negative words than the shorter ones.

3.3.2 Method 2: Sentiment score normalization by line's length

The second method arises from the first one normalizing the sentiment score based on the length of the line. For this reason, the difference of positive and negative words will be divided by the word count of the corresponding line.

$$SentimentScore = \frac{\sum(positiveword) - \sum(negativewords)}{Linewordcount}$$

The produced result takes into account the length of the corresponding line. Nevertheless, the word count value of the line could be small and hard to differentiate the result. For this reason, a predefined multiplier could be used in order to leverage results.

3.3.3 Method 3: Sentiment score calculation using the ratio of positive and negative word count

The last custom method for sentiment score calculation that was implemented takes into consideration the polarity of positive and negative words.

$$SentimentScore = \frac{\sum(positiveword)}{\sum(negativewords) + 1}$$

This method tends to be a more balanced way of calculation between interpretation and normalization. One of the main disadvantages of this method is that the determination of neutral sentiments could be harder because the denominator is incremented by one (+1).

3.3.4 Method 4: TextBlob library usage for sentiment score calculation

Apart from the custom-implemented methods for sentiment score calculation, a ready-to-use Python library was experimentally employed. TextBlob library was selected due to its simplicity and easy-to-use value. Sentiment score calculation by TextBlob returns a tuple of form (polarity, subjectivity). The polarity has values in a range between $[-1.0, 1.0]$, while subjectivity fluctuates within the range $[0.0, 1.0]$, where value 0.0 is very objective and value 1.0 is very subjective. Since the purpose of this study is to analyse the emotion behind each script, the subjectivity factor seems to be out of scope. For this reason, in the specific implementation, polarity is the sole factor taken into consideration in order to calculate the sentiment score.

4 Evaluation Results

Eight different scripts were used in order to evaluate the present implementation. One of them is custom-created only for evaluation purposes, whereas all the rest are real ones. The list of words (stop words, negative words and positive words) was created based on personal experience. Of course, the scripts and the lists of words can change at any execution, as they can be configured in the configuration file. The aim of this section is to demonstrate the specific implementation and evaluate produced results.

4.1 Preparation

In the present implementation, one of the most important components is the configuration file. It is crucial because, among all other useful data, it contains the lists of positive and negative words. These lists are important for this particular implementation, as they are used in custom-made methods for sentiment score calculation. They contain data personal experience regarding this subject, along with the research conducted. At this point, it is worth mentioning that these lists contain words that can be used only against tokenized content (notably words), as is the case here. Neither in these lists, nor in the custom-implemented methods for sentiment word calculation can phrases be used. Furthermore, a stop-word list exists in order to remove words that are not useful in determining sentiment.

In order to evaluate this implementation, eight different scripts were used:

1. A custom one named as sample.txt, which was used only to evaluate the functioning of the implementation.
2. Six random movies were selected, for which no previous knowledge had been obtained. Those were “Joker”, “A few good men”, “Candle to water”, “Unknown”, “Next” and “Valkyrie”.
3. Finally, the script of a movie that had been criticized for its content was selected. This movie is “Blazing Saddles” and it was previously mentioned on §2.2.2.3.

As a result, the configuration file used was the following one:

```
[basic-config]
stop_words = 'white'
positive_words = 'black', 'gay', 'boy', 'love', 'i', 'he', she, 'man', 'men', 'girl', 'girls', 'go', 'come', 'hi'
negative_words = 'nigger', 'bitch', 'tranny'
movies_list=movie-joker,movie-blazing_saddles,movie-a_few_good_men,movie-candle_to_water,movie-unknown,movie-next,movie-valkyrie,movie-sample
visualizations_export=VisualizationsExport.xlsx
[tf-idf]
train_size=0.80
test_size=0.20
random_state=101
min_df=0.0001
max_df=0.95
ngram_range_start=1
ngram_range_stop=3
metrics_export=TF-IDF-metrics.txt
[movie-joker]
movie_name = JOKER
```

```
script_path = JOKER.txt
[movie-blazing_saddles]
movie_name = BLAZING SADDLES
script_path = BLAZING_SADDLES.txt
[movie-a_few_good_men]
movie_name = A FEW GOOD MEN
script_path = A_FEW GOOD_MEN.txt
[movie-candle_to_water]
movie_name = CANDLE TO WATER
script_path = CANDLE_TO_WATER.txt
[movie-unknown]
movie_name = UNKNOWN
script_path = UNKNOWN.txt
[movie-next]
movie_name = NEXT
script_path = NEXT.txt
[movie-valkyrie]
movie_name = VALKYRIE
script_path = VALKYRIE.txt
[movie-sample]
movie_name = sample
script_path = sample.txt
```

As it may be observed, the basic configuration was highlighted in yellow color. In this section, the three lists mentioned above (stop words, positive words and negative words), the list of movies to be examined and the name of the Excel file containing the results are included. In the following section lie the parameters used to implement the TF-IDF method along with the name of the file that the metrics will be stored (all of them highlighted in green color). Data related to movies are highlighted in light blue color. Subsequently, the sample movie used only for evaluation purposes is presented. It simply contains some lines with words existing both in negative and positive lists.

I love burger fellow gay nigger

I love burger fellow gay black waitress

my name is black panther

i love nigger guy

hi all

black power

My name is Tester

go away tranny

we need to kick off all tranny nigger boys from neighborhood

I love cars

Go go Chelsea

Stupid tranny nigger

4.2 Execution

Having successfully executed the Python application, through the configuration described in the previous section, several PNG and txt files are produced (two PNG files and two txt files for every movie) plus an Excel file with as many sheets as the configured movie scripts as well as one txt file containing TF-IDF algorithm metrics. In this section, the results for each movie will be separately examined.

4.2.1 Movie Script: sample

As shown on Figure 5, the first two methods for sentiment score calculation (SC1 and SC2) produce equal results, whereas the third method (SC_3) yields different results. The fourth method, which uses a library (TextBlob), provides a different perspective and this is an expected behavior as it uses a different approach. Figure 6 displays a word cloud with nothing mentioned, as this script is used only for demonstration purposes. Results presented on Table 2 are equal to the ones presented on Figure 5. Figure 7 presents the TF-IDF algorithm classification report and Figure 8 presents TF-IDF algorithm confusion matrix. Moreover, Figure 9 presents TF-IDF metrics and these results are equal to the ones presented on Figure 7. Using the custom-made script, it was possible to verify the functioning of the implementation as expected, because a small number of lines and words still remained to be examined.

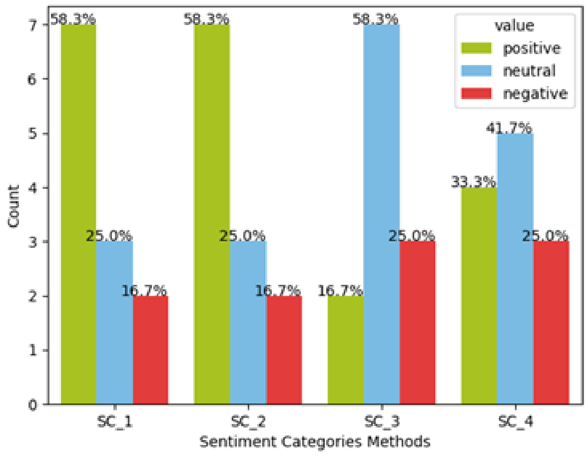

Figure 5, Sample movie script visualizations

Figure 6, Sample movie script word cloud

	SC_1 Count	**SC_2 Count**	**SC_3 Count**	**SC_4 Count**	**SC_1 %**	**SC_2 %**	**SC_3 %**	**SC_4 %**
negative	2	2	3	3	16,66667	16,66667	25	25
neutral	3	3	7	5	25	25	58,33333	41,66667
positive	7	7	2	4	58,33333	58,33333	16,66667	33,33333

Table 2, Sample movie script results

```
Method: SC_1
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00         1
    positive       0.67      1.00      0.80         2

    accuracy                           0.67         3
   macro avg       0.33      0.50      0.40         3
weighted avg       0.44      0.67      0.53         3

Method: SC_2
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00         1
    positive       0.67      1.00      0.80         2

    accuracy                           0.67         3
   macro avg       0.33      0.50      0.40         3
weighted avg       0.44      0.67      0.53         3

Method: SC_3
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00         1
     neutral       0.67      1.00      0.80         2

    accuracy                           0.67         3
   macro avg       0.33      0.50      0.40         3
weighted avg       0.44      0.67      0.53         3

Method: SC_4
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00         1
     neutral       0.50      1.00      0.67         1
    positive       1.00      1.00      1.00         1

    accuracy                           0.67         3
   macro avg       0.50      0.67      0.56         3
weighted avg       0.50      0.67      0.56         3
```

Figure 7, Sample movie TF-IDF classification report

```
Method: SC_1
[[  0   1   0]
 [  0 966   0]
 [  0  22   0]]

Method: SC_2
[[  0   1   0]
 [  0 966   0]
 [  0  22   0]]

Method: SC_3
[[960   0]
 [ 29   0]]

Method: SC_4
[[  0  79   0]
 [  0 798   0]
 [  0 108   4]]
```

Figure 8, Sample movie TF-IDF confusion matrix

Movie: sample

Method	Number of features	Accuracy	Precision	Recall	F1
SC_1	34	66.67	44.44	66.67	53.33
SC_2	34	66.67	44.44	66.67	53.33
SC_3	34	66.67	44.44	66.67	53.33
SC_4	34	66.67	50.0	66.67	55.56

Figure 9, Sample movie TF-IDF metrics

4.2.2 Movie Script: *Joker*

Having examined the produced results, almost everything appeared as expected, except for the results for SC_3, as presented on Figure 10. The results for this method are very different from the ones calculated using other methods. Therefore, the setup of this method is subject to further investigation. Furthermore, the word cloud presented on Figure 11 is very close to the subject of the movie script and there is nothing peculiar about it. TF-IDF metrics for this movie, presented on Figure 12 and Figure 14, show that custom sentiment score methods produce better results. Although the third sentiment score method produces better TF-IDF metrics compared to other methods, it requires further investigation.

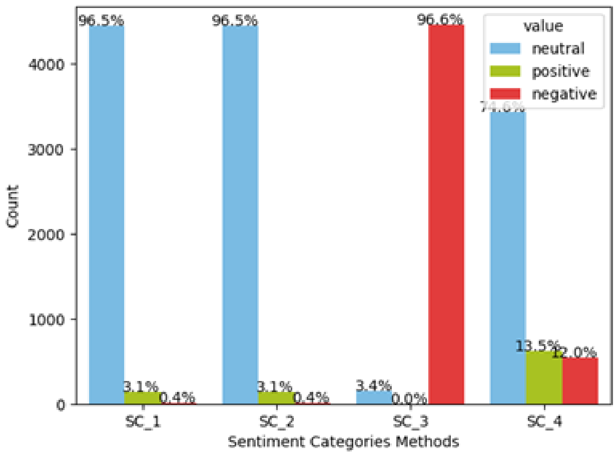

Figure 10, Joker script visualizations

Figure 11, Joker script word cloud

	SC_1 Count	SC_2 Count	SC_3 Count	SC_4 Count	SC_1 %	SC_2 %	SC_3 %	SC_4 %
negative	19	19	4456	552	0,411879	0,411879	96,59657	11,96618
neutral	4450	4450	155	3439	96,46651	96,46651	3,360069	74,55018
positive	144	144	2	622	3,121613	3,121613	0,043356	13,48363

Table 3, Joker script results

Method: SC_1

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	4
neutral	0.96	1.00	0.98	887
positive	0.00	0.00	0.00	32
accuracy			0.96	923
macro avg	0.32	0.33	0.33	923
weighted avg	0.92	0.96	0.94	923

Method: SC_2

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	4
neutral	0.96	1.00	0.98	887
positive	0.00	0.00	0.00	32
accuracy			0.96	923
macro avg	0.32	0.33	0.33	923
weighted avg	0.92	0.96	0.94	923

Method: SC_3

	precision	recall	f1-score	support
negative	0.96	1.00	0.98	888
neutral	0.00	0.00	0.00	35
accuracy			0.96	923
macro avg	0.48	0.50	0.49	923
weighted avg	0.93	0.96	0.94	923

Method: SC_4

	precision	recall	f1-score	support
negative	1.00	0.06	0.12	111
neutral	0.77	1.00	0.87	684
positive	1.00	0.17	0.29	128
accuracy			0.77	923
macro avg	0.92	0.41	0.43	923
weighted avg	0.83	0.77	0.70	923

Figure 12, Joker TF-IDF classification report

```
Method: SC_1
[[  0   4   0]
 [  0 887   0]
 [  0  32   0]]

Method: SC_2
[[  0   4   0]
 [  0 887   0]
 [  0  32   0]]

Method: SC_3
[[888   0]
 [ 35   0]]

Method: SC_4
[[  7 104   0]
 [  0 684   0]
 [  0 106  22]]
```

Figure 13, Joker TF-IDF confusion matrix

Movie: JOKER

Method	Number of features	Accuracy	Precision	Recall	F1
SC_1	12420	96.1	92.35	96.1	94.19
SC_2	12420	96.1	92.35	96.1	94.19
SC_3	12420	96.21	92.56	96.21	94.35
SC_4	12420	77.25	82.59	77.25	69.74

Figure 14, Joker TF-IDF metrics

4.2.3 Movie Script: *Blazing Saddles*

This movie script was used in the evaluation conducted concerning a movie that has been criticized for its content. All the results presented on Figure 15 are as expected, apart from the ones based on the third method of sentiment score calculation. As these results seem to be quite unrefined, this method remains to be further examined. Additionally, the results presented on Figure 16 meet expectations, as the words presented on the word cloud are quite similar to the plot of the movie. Figure 17 and Figure 19 show that custom sentiment score methods produce better results. Although the third sentiment score method produces better TF-IDF metrics compared to other methods, it requires further investigation.

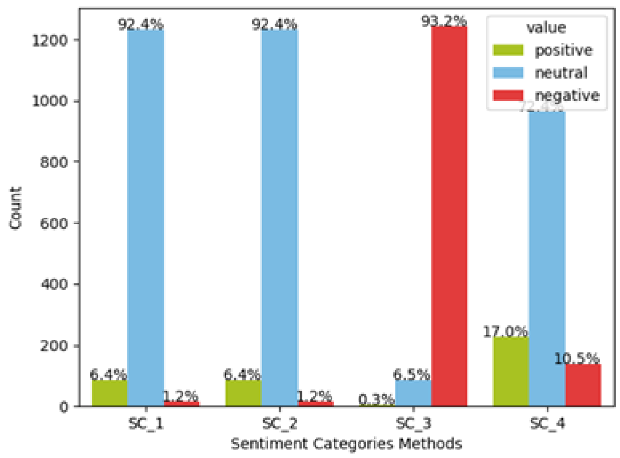

Figure 15, Blazing Saddles script visualizations

Figure 16, Blazing Saddles script word cloud

	SC_1 Count	SC_2 Count	SC_3 Count	SC_4 Count	SC_1 %	SC_2 %	SC_3 %	SC_4 %
negative	16	16	1242	140	1,201201	1,201201	93,24324	10,51051
neutral	1231	1231	86	965	92,41742	92,41742	6,456456	72,44745
positive	85	85	4	227	6,381381	6,381381	0,3003	17,04204

Table 4, Blazing Saddles script results

```
Method: SC_1
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00         7
     neutral       0.91      1.00      0.95       244
    positive       0.00      0.00      0.00        16

    accuracy                           0.91       267
   macro avg       0.30      0.33      0.32       267
weighted avg       0.84      0.91      0.87       267

Method: SC_2
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00         7
     neutral       0.91      1.00      0.95       244
    positive       0.00      0.00      0.00        16

    accuracy                           0.91       267
   macro avg       0.30      0.33      0.32       267
weighted avg       0.84      0.91      0.87       267

Method: SC_3
              precision    recall  f1-score   support

    negative       0.94      1.00      0.97       250
     neutral       0.00      0.00      0.00        15
    positive       0.00      0.00      0.00         2

    accuracy                           0.94       267
   macro avg       0.31      0.33      0.32       267
weighted avg       0.88      0.94      0.91       267

Method: SC_4
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00        33
     neutral       0.71      1.00      0.83       187
    positive       1.00      0.09      0.16        47

    accuracy                           0.72       267
   macro avg       0.57      0.36      0.33       267
weighted avg       0.67      0.72      0.61       267
```

Figure 17, Blazing Saddles TF-IDF classification report

```
Method: SC_1
[[  0   7   0]
 [  0 244   0]
 [  0  16   0]]

Method: SC_2
[[  0   7   0]
 [  0 244   0]
 [  0  16   0]]

Method: SC_3
[[250   0   0]
 [ 15   0   0]
 [  2   0   0]]

Method: SC_4
[[  0  33   0]
 [  0 187   0]
 [  0  43   4]]
```

Figure 18, Blazing Saddles TF-IDF confusion matrix

Movie: BLAZING SADDLES

Method	Number of features	Accuracy	Precision	Recall	F1
SC_1	2898	91.39	83.51	91.39	87.27
SC_2	2898	91.39	83.51	91.39	87.27
SC_3	2898	93.63	87.67	93.63	90.55
SC_4	2898	71.54	67.4	71.54	60.97

Figure 19, Blazing Saddles TF-IDF metrics

4.2.4 Movie Script: *A Few Good Men*

TF-IDF metrics for this movie, presented ov Figure 22 and Figure 24, show that custom sentiment score methods produce better results, especially the first two sentiment score methods.

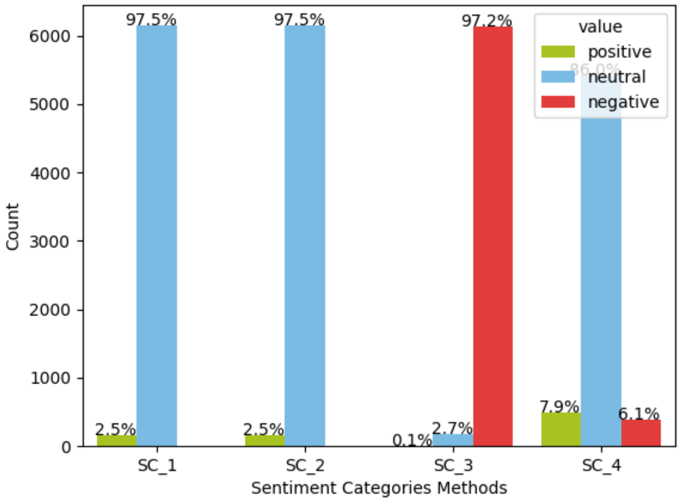

Figure 20, A Few Good Men script visualizations

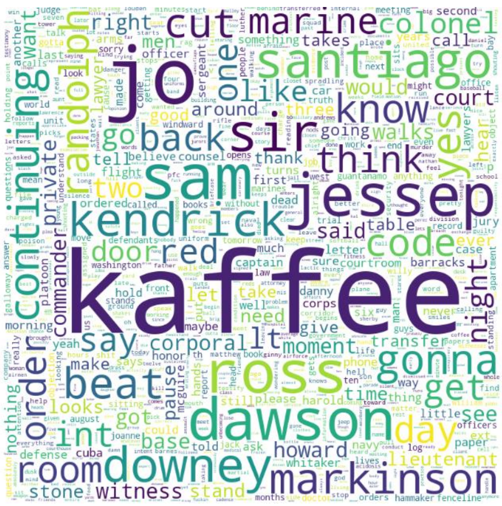

Figure 21, A Few Good Men script word cloud

	SC_1 Count	SC_2 Count	SC_3 Count	SC_4 Count	SC_1 %	SC_2 %	SC_3 %	SC_4 %
negative			6128	383			97.22354	6.076472
neutral	6145	6145	171	5420	97.49326	97.49326	2.712994	85.9908
positive	158	158	4	500	2.506743	2.506743	0.063462	7.93273

Table 5, A Few Good Men script results

```
Method: SC_1
              precision    recall  f1-score   support

     neutral       0.97      1.00      0.99      1227
    positive       0.00      0.00      0.00        34

    accuracy                           0.97      1261
   macro avg       0.49      0.50      0.49      1261
weighted avg       0.95      0.97      0.96      1261

Method: SC_2
              precision    recall  f1-score   support

     neutral       0.97      1.00      0.99      1227
    positive       0.00      0.00      0.00        34

    accuracy                           0.97      1261
   macro avg       0.49      0.50      0.49      1261
weighted avg       0.95      0.97      0.96      1261

Method: SC_3
              precision    recall  f1-score   support

    negative       0.97      1.00      0.99      1225
     neutral       0.00      0.00      0.00        35
    positive       0.00      0.00      0.00         1

    accuracy                           0.97      1261
   macro avg       0.32      0.33      0.33      1261
weighted avg       0.94      0.97      0.96      1261

Method: SC_4
              precision    recall  f1-score   support

    negative       1.00      0.03      0.06        64
     neutral       0.87      1.00      0.93      1095
    positive       1.00      0.06      0.11       102

    accuracy                           0.87      1261
   macro avg       0.96      0.36      0.37      1261
weighted avg       0.89      0.87      0.82      1261
```

Figure 22, A Few Good Men TF-IDF classification report

```
Method: SC_1
[[1227    0]
 [  34    0]]

Method: SC_2
[[1227    0]
 [  34    0]]

Method: SC_3
[[1225    0    0]
 [  35    0    0]
 [   1    0    0]]

Method: SC_4
[[   2   62    0]
 [   0 1095    0]
 [   0   96    6]]
```

Figure 23, A Few Good Men TF-IDF confusion matrix

Movie: A FEW GOOD MEN

Method	Number of features	Accuracy	Precision	Recall	F1
SC_1	10344	97.3	94.68	97.3	95.97
SC_2	10344	97.3	94.68	97.3	95.97
SC_3	10344	97.15	94.37	97.15	95.74
SC_4	10344	87.47	89.05	87.47	82.2

Figure 24, A Few Good Men TF-IDF metrics

4.2.5 Movie Script: *Candle to Water*

TF-IDF metrics for this movie, presented on Figure 27 and Figure 29, show that custom sentiment score methods yield better results, as is obvious through the first two sentiment score methods.

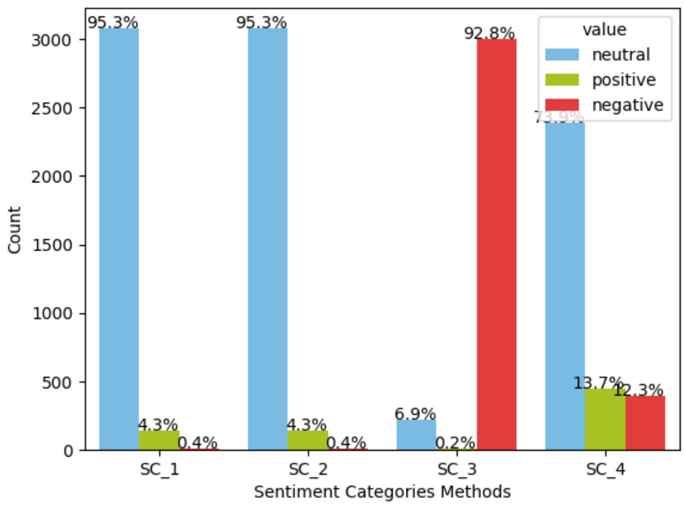

Figure 25, Candle to Water script visualizations

Figure 26, Candle to Water script word cloud

	SC_1 Count	SC_2 Count	SC_3 Count	SC_4 Count	SC_1 %	SC_2 %	SC_3 %	SC_4 %
negative	14	14	2998	398	0.433437	0.433437	92.81734	12.32198
neutral	3077	3077	224	2388	95.26316	95.26316	6.934985	73.93189
positive	139	139	8	444	4.303406	4.303406	0.247678	13.74613

Table 6, Candle to Water script results

Method: SC_1

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	2
neutral	0.95	1.00	0.97	612
positive	0.00	0.00	0.00	32
accuracy			0.95	646
macro avg	0.32	0.33	0.32	646
weighted avg	0.90	0.95	0.92	646

Method: SC_2

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	2
neutral	0.95	1.00	0.97	612
positive	0.00	0.00	0.00	32
accuracy			0.95	646
macro avg	0.32	0.33	0.32	646
weighted avg	0.90	0.95	0.92	646

Method: SC_3

	precision	recall	f1-score	support
negative	0.93	1.00	0.96	596
neutral	1.00	0.04	0.08	49
positive	0.00	0.00	0.00	1
accuracy			0.93	646
macro avg	0.64	0.35	0.35	646
weighted avg	0.93	0.93	0.89	646

Method: SC_4

	precision	recall	f1-score	support
negative	1.00	0.15	0.26	79
neutral	0.77	1.00	0.87	481
positive	1.00	0.12	0.21	86
accuracy			0.78	646
macro avg	0.92	0.42	0.45	646
weighted avg	0.83	0.78	0.71	646

Figure 27, Candle to Water TF-IDF classification report

```
Method: SC_1
[[  0   2   0]
 [  0 612   0]
 [  0  32   0]]

Method: SC_2
[[  0   2   0]
 [  0 612   0]
 [  0  32   0]]

Method: SC_3
[[596   0   0]
 [ 47   2   0]
 [  1   0   0]]

Method: SC_4
[[ 12  67   0]
 [  0 481   0]
 [  0  76  10]]
```

Figure 28, Candle to Water TF-IDF confusion matrix

Movie: CANDLE TO WATER

Method	Number of features	Accuracy	Precision	Recall	F1
SC_1	9746	94.74	89.75	94.74	92.18
SC_2	9746	94.74	89.75	94.74	92.18
SC_3	9746	92.57	92.97	92.57	89.28
SC_4	9746	77.86	82.94	77.86	70.82

Figure 29, Candle to Water TF-IDF metrics

4.2.6 Movie Script: *Next*

TF-IDF metrics for this movie, presented on Figure 32 and Figure 34, show that custom sentiment score methods produce better results, as particularly shown through the first two sentiment score methods.

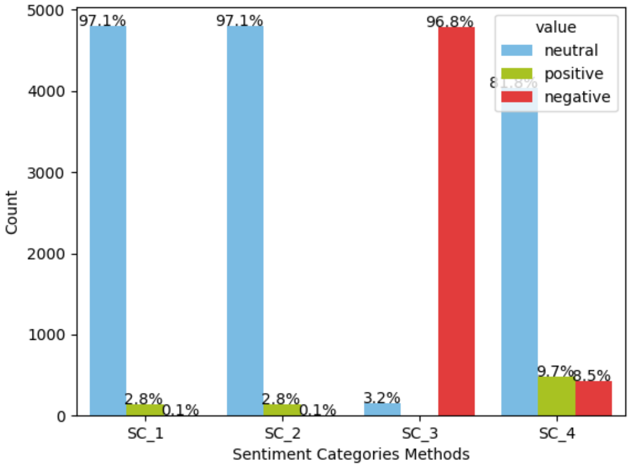

Figure 30, Next script visualizations

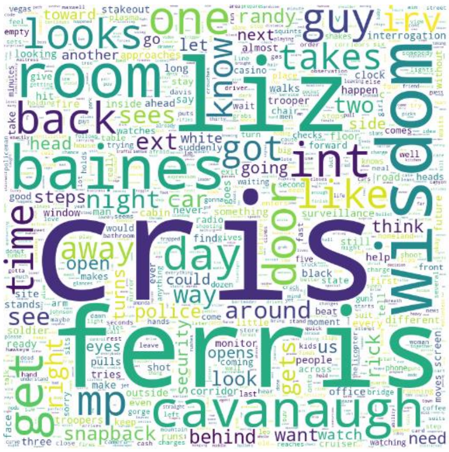

Figure 31, Next script word cloud

	SC_1 Count	SC_2 Count	SC_3 Count	SC_4 Count	SC_1 %	SC_2 %	SC_3 %	SC_4 %
negative	5	5	4783	419	0.101194	0.101194	96.80227	8.480065
neutral	4799	4799	158	4041	97.12609	97.12609	3.197733	81.78506
positive	137	137		481	2.772718	2.772718		9.734871

Table 7, Next script results

```
Method: SC_1
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00         1
     neutral       0.98      1.00      0.99       966
    positive       0.00      0.00      0.00        22

    accuracy                           0.98       989
   macro avg       0.33      0.33      0.33       989
weighted avg       0.95      0.98      0.97       989

Method: SC_2
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00         1
     neutral       0.98      1.00      0.99       966
    positive       0.00      0.00      0.00        22

    accuracy                           0.98       989
   macro avg       0.33      0.33      0.33       989
weighted avg       0.95      0.98      0.97       989

Method: SC_3
              precision    recall  f1-score   support

    negative       0.97      1.00      0.99       960
     neutral       0.00      0.00      0.00        29

    accuracy                           0.97       989
   macro avg       0.49      0.50      0.49       989
weighted avg       0.94      0.97      0.96       989

Method: SC_4
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00        79
     neutral       0.81      1.00      0.90       798
    positive       1.00      0.04      0.07       112

    accuracy                           0.81       989
   macro avg       0.60      0.35      0.32       989
weighted avg       0.77      0.81      0.73       989
```

Figure 32, Next TF-IDF classification report

```
Method: SC_1
[[  0   1   0]
 [  0 966   0]
 [  0  22   0]]

Method: SC_2
[[  0   1   0]
 [  0 966   0]
 [  0  22   0]]

Method: SC_3
[[960   0]
 [ 29   0]]

Method: SC_4
[[  0  79   0]
 [  0 798   0]
 [  0 108   4]]
```

Figure 33, Next TF-IDF confusion matrix

Movie: NEXT

Method	Number of features	Accuracy	Precision	Recall	F1
SC_1	12211	97.67	95.4	97.67	96.53
SC_2	12211	97.67	95.4	97.67	96.53
SC_3	12211	97.07	94.22	97.07	95.62
SC_4	12211	81.09	76.69	81.09	73.01

Figure 34, Next TF-IDF metrics

4.2.7 Movie Script: *Unknown*

TF-IDF metrics for this movie, presented on Figure 37 and Figure 39, show that custom sentiment score methods produce better results, as shown by the first two sentiment score methods.

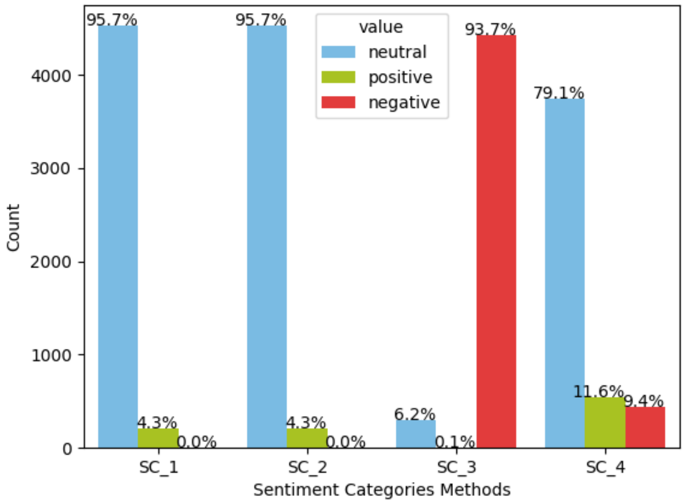

Figure 35, Unknown script visualizations

Figure 36, Unknown script word cloud

	SC_1 Count	SC_2 Count	SC_3 Count	SC_4 Count	SC_1 %	SC_2 %	SC_3 %	SC_4 %
negative	1	1	4433	443	0.021133	0.021133	93.68132	9.361792
neutral	4527	4527	292	3742	95.66779	95.66779	6.170752	79.07861
positive	204	204	7	547	4.311074	4.311074	0.147929	11.55959

Table 8, Unknown script results

```
Method: SC_1
              precision    recall  f1-score   support

     neutral       0.96      1.00      0.98       911
    positive       0.00      0.00      0.00        36

    accuracy                           0.96       947
   macro avg       0.48      0.50      0.49       947
weighted avg       0.93      0.96      0.94       947

Method: SC_2
              precision    recall  f1-score   support

     neutral       0.96      1.00      0.98       911
    positive       0.00      0.00      0.00        36

    accuracy                           0.96       947
   macro avg       0.48      0.50      0.49       947
weighted avg       0.93      0.96      0.94       947

Method: SC_3
              precision    recall  f1-score   support

    negative       0.95      1.00      0.97       897
     neutral       0.00      0.00      0.00        49
    positive       0.00      0.00      0.00         1

    accuracy                           0.95       947
   macro avg       0.32      0.33      0.32       947
weighted avg       0.90      0.95      0.92       947

Method: SC_4
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00        97
     neutral       0.80      1.00      0.89       757
    positive       1.00      0.05      0.10        93

    accuracy                           0.80       947
   macro avg       0.60      0.35      0.33       947
weighted avg       0.74      0.80      0.72       947
```

Figure 37, Unknown TF-IDF classification report

```
Method: SC_1
[[911   0]
 [ 36   0]]

Method: SC_2
[[911   0]
 [ 36   0]]

Method: SC_3
[[897   0   0]
 [ 49   0   0]
 [  1   0   0]]

Method: SC_4
[[  0  97   0]
 [  0 757   0]
 [  0  88   5]]
```

Figure 38, Unknown TF-IDF confusion matrix

Movie: UNKNOWN

Method	Number of features	Accuracy	Precision	Recall	F1
SC_1	15703	96.2	92.54	96.2	94.33
SC_2	15703	96.2	92.54	96.2	94.33
SC_3	15703	94.72	89.72	94.72	92.15
SC_4	15703	80.46	74.06	80.46	72.23

Figure 39, Unknown TF-IDF metrics

4.2.8 Movie Script: *Valkyrie*

TF-IDF metrics for this movie, presented on Figure 42 and Figure 44, show that custom sentiment score methods produce better results.

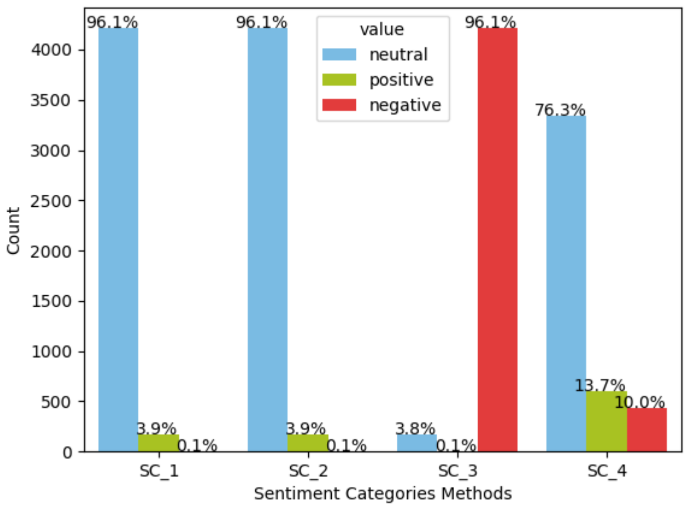

Figure 40, Valkyrie script visualizations

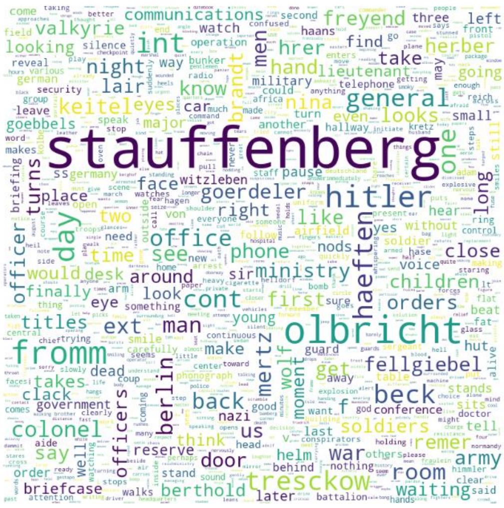

Figure 41, Valkyrie script word cloud

	SC_1 Count	SC_2 Count	SC_3 Count	SC_4 Count	SC_1 %	SC_2 %	SC_3 %	SC_4 %
negative	3	3	4213	439	0.068431	0.068431	96.09945	10.01369
neutral	4211	4211	165	3345	96.05383	96.05383	3.763686	76.30018
positive	170	170	6	600	3.877737	3.877737	0.136861	13.68613

Table 9, Valkyrie script results

```
Method: SC_1
              precision    recall  f1-score   support

     neutral       0.96      1.00      0.98       842
    positive       0.00      0.00      0.00        35

    accuracy                           0.96       877
   macro avg       0.48      0.50      0.49       877
weighted avg       0.92      0.96      0.94       877

Method: SC_2
              precision    recall  f1-score   support

     neutral       0.96      1.00      0.98       842
    positive       0.00      0.00      0.00        35

    accuracy                           0.96       877
   macro avg       0.48      0.50      0.49       877
weighted avg       0.92      0.96      0.94       877

Method: SC_3
              precision    recall  f1-score   support

    negative       0.96      1.00      0.98       842
     neutral       0.00      0.00      0.00        34
    positive       0.00      0.00      0.00         1

    accuracy                           0.96       877
   macro avg       0.32      0.33      0.33       877
weighted avg       0.92      0.96      0.94       877

Method: SC_4
              precision    recall  f1-score   support

    negative       0.00      0.00      0.00        96
     neutral       0.77      1.00      0.87       671
    positive       1.00      0.09      0.17       110

    accuracy                           0.78       877
   macro avg       0.59      0.36      0.35       877
weighted avg       0.72      0.78      0.69       877
```

Figure 42, Valkyrie TF-IDF classification report

```
Method: SC_1
[[842   0]
 [ 35   0]]

Method: SC_2
[[842   0]
 [ 35   0]]

Method: SC_3
[[842   0   0]
 [ 34   0   0]
 [  1   0   0]]

Method: SC_4
[[  0  96   0]
 [  0 671   0]
 [  0 100  10]]
```

Figure 43, Valkyrie TF-IDF confusion matrix

Movie: VALKYRIE

Method	Number of features	Accuracy	Precision	Recall	F1
SC_1	12089	96.01	92.18	96.01	94.05
SC_2	12089	96.01	92.18	96.01	94.05
SC_3	12089	96.01	92.18	96.01	94.05
SC_4	12089	77.65	71.76	77.65	68.85

Figure 44, Valkyrie TF-IDF metrics

4.3 Summary

First of all, it should be noted that the present implementation works well either with a sample script or with real ones. For this reason, an evaluation based on a custom-made script was initially executed. Furthermore, results in visualizations, figures and tables are equal, although different methods for calculating them were used in the developed code. This is also proof of the proper functioning of this implementation. In addition, figures presenting word clouds seem to be well defined, as they display words common to the subject and the plot of each movie. Finally, the most important piece of evidence that the implementation works as designed are the results for *Blazing Saddles* movie script.

TF-IDF algorithm was applied to all movies. As is obvious, both classification report and metrics produce equal results. This is an expected occurrence through which the functioning of the implementation can be validated. TF-IDF metrics contribute to the evaluation of the implementation as well as the identification of the most appropriate sentiment score method to produce better results.

The only drawback that was noticed and needs to be further examined are the results for the third method for sentiment score calculation. As previously mentioned, these results seem to be unrefined compared to the rest and further work needs to be conducted. For this reason, TF-IDF metrics produced for that method won't be taken into consideration.

5 Conclusions

One characteristic of the human race that has existed so far is the categorization of people according to some characteristics of theirs, such as skin color. This type of distinction has been common for all societies throughout history. Most of the times, this categorization is negative and leads to violence. It can be expressed in many ways and through different means, such as books, articles, movies, etc. Nowadays, this distinction is mainly expressed through racism and homophobia. Those are two attributes that can be expressed in different ways and are sometimes found in art. Some indicative examples are movies or plays where racism and homophobic content exist. Sometimes it is difficult to discover movies or plays with such content, whereas in other cases they need to be examined more carefully. Should there be any way to find out which movies or plays contain such content based on their script, it would be very useful in order for the audience to be notified accordingly.

The evolution of technology has provided many tools and frameworks which can be used in every aspect of everyday life, from chatting online with friends to identifying customers’ preferences through their comments on social media. Especially the latter resource of technology is the subject of sentiment analysis. Sentiment analysis uses NLP to determine whether data are positive, negative or neutral. SA is often performed on textual data in order to understand customer preferences. It is mainly used by businesses on social media to detect sentiment through social data. Sentiment exists in text and can be either positive, negative or neutral.

Programming languages are a part of every aspect of technology from mobile phones to custom reports for e-shops growth. There are plenty of languages that someone can use to address the problem they are working on. For instance, a website can be developed using either pure HTML or JavaScript. Undoubtedly, using JavaScript on a website will have many more capabilities compared to one that will be developed using solely HTML. One of the fastest growing languages with many capabilities is Python. It is used very often by developers due to the unique features it provides. One of them is the majority of available libraries covering every possible area. As a programming language, it has its own limitations and it is not ideal for every problem, but it is widely accepted that Python is an ideal programming language for data manipulation and analysis problems.

To come back to the main topic of this thesis, Python programming language was used in order to identify racism and homophobic content in movie scripts and plays. This programming language is an excellent choice as there are many libraries to use and it is extremely handful when data manipulation issues arise. Four different methods for sentiment score calculation were implemented. The first three methods use custom implementation and the fourth one uses a predefined library. At this point, it should be noted that three different lists of words should appear within the custom-implemented methods. These word lists will be used within the custom-implemented methods. It is worth mentioning that sentiment score is calculated separately for each script, parsing it line by line. For this reason, each line should be tokenized prior to applying the custom-implemented methods for sentiment score calculation. Results from sentiment score calculation are presented in figures and tables. Furthermore, a word cloud figure was created for each movie script.

Evaluating produced results, it became evident that only the first two custom-implemented methods for sentiment score calculation produce acceptable results. The third method, sentiment score calculation using the ratio of positive and negative word count, needs to be tuned as it yields some unrefined results. One significant advantage of the three custom methods implemented is that they use word lists which were personally defined. These lists can be specifically configured and contain words which are subject-related, in this case, racism and homophobic content. The fourth method uses a publicly available library. This library can be replaced with any other that could possibly be useful. The disadvantage of using a predefined library without word lists is that it is too general and not specific to the issue that is examined. Certainly, using such a library can eliminate development time and minimize the risk of using a custom-implemented code.

In the present application, the percentage of negative, positive and neutral sentiments is simply identified within a movie script. For every single movie, the percentage of each sentiment and the number of lines per sentiment are presented. These results are produced by the calculations the application has made without any training or evaluation. User reviews and movie ranking could be used for this purpose. Furthermore, the third custom-implemented method should be improved as it can play a significant role in sentiment score calculation. An optional feature could be web crawling of movie scripts and plays. A user could provide a list of movies and plays and the application should be able to download these scripts instead of the user providing them in a predefined path. Moreover, in a future release of the application, it might be possible to experiment with different publicly available libraries. This experiment will help to identify which library is the most suitable one. Last but not least, figures and tables produced through the application could be improved by combining them all into one. All these improvements can take place in the future in order to further improve the application developed.

Using the TF-IDF algorithm, it was concluded that the custom-implemented methods produce better results, relying upon the classification report and metrics produced. Metrics produced for each sentiment score method are very helpful for the evaluation that was conducted. Several text files for each movie displaying results from applying TF-IDF algorithm were created. One future enhancement could be the creation of figures displaying those results. Moreover, as a future enhancement, produced metrics could be saved in an Excel file, so as to be processed more easily. More machine learning algorithms can be added to a future release, especially deep learning models that are extensively used today. Due to the methodology that was followed, it could be easier to adopt new machine learning models and incorporate them into the current application.

6 APPENDIX

6.1 Code Analysis

The present implementation consists of the following files:

- MyApplicationProperties.txt: it is the configuration file of the implementation and should exist in the same folder with the Python files.
- Application.py: contains the logic behind this implementation and performs the processing based on the methodology presented above.
- Configuration.py: it contains classes used to hold the values of the aforementioned configuration file.
- Model.py: it contains classes for the model of this application. Currently, it only has one class for the TF-IDF approach.

The name and location of the text file is hard coded inside the Application.py Python file. The version of Python used is 3.9.2. Below is an analysis of the code developed in Application.py file in order to implement the above presented methodology.

First, the required libraries will be imported.

```
# Import Libraries
import os
import warnings
```

```python
import configparser

import numpy as np
import openpyxl
from openpyxl import Workbook
from prettytable import PrettyTable
from Configuration import BasicConfiguration, TfIdfConfiguration
from Model import TfIdfModel
# Preprocessing
import nltk
import pandas as pd
from pandas import ExcelWriter
import re
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from textblob import TextBlob
from wordcloud import WordCloud
# Plt
import matplotlib.pyplot as plt
import seaborn as sns
# To sort dictionary values
import operator
# TF-IDF
import sklearn.model_selection as model_selection
from sklearn.pipeline import Pipeline
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import metrics
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
```

Moving forward, some initializations that are common for the whole Python file will be made:

- Warnings will be ignored, as they have been checked and they might be confusing.

- The name of the configuration file is MyApplicationProperties.txt and exists in the same location as the Python file. Configuration file consists of at least two sections:
    - A basic configuration section appearing in every configuration file, as it is hard coded, having the same name. This section contains the basic parameters.
    - A configuration section for TF-IDF algorithm, which also has the same name in every configuration file as it is hard coded. This section contains information for TF-IDF algorithm parameters.
    - For each movie that this implementation will process, a separate section containing information and parameters exists.
- The Porter Stemming algorithm will be employed.
- Stop words will appear in English.
- Furthermore, the output display will be expanded in order to be able to see more columns of a Pandas DataFrame.

At this point, it is worth mentioning that Pandas DataFrame is the main data structure that is going to be used in this application. A Pandas DataFrame contains two-dimensional, size-mutable, potentially heterogeneous tabular data (pandas development team, n.d.).

```
warnings.filterwarnings("ignore")

CONFIG_FILE_PATH = r'MyApplicationProperties.txt'
BASIC_CONFIGURATION_SECTION = 'basic-config'
TF_IDF_CONFIGURATION_SECTION = 'tf-idf'

st = PorterStemmer()
set(stopwords.words('english'))

pd.set_option('display.width', 4000)
pd.set_option('display.max_columns', 100)
```

Two functions for printing either dataframe or data as shown below have been created.

```
# Print Dataframe
def print_dataframe(df):
    print("\nData Frame: ")
    print(df.head())


# Print Data
def print_data(data):
    print("\nData: ")
    print(data)
```

Another function is responsible for reading predefined configuration files and returning a configuration file parser.

```
def read_configuration():
    config_parser = configparser.RawConfigParser()
    config_parser.read(CONFIG_FILE_PATH)

    return config_parser
```

One of the most essential functions implemented is the one that loads a movie script file on a Pandas DataFrame.

```
# Load Script
def load_movie_script(file_name):
    df_script = pd.read_table(file_name, header=None)
    df_script.columns = ["Dialogue"]

    return df_script;
```

A very important task in the present implementation is data pre-processing. The function presented below is extremely significant as it clears content that is not useful in determining sentiment. At this point, it should be noted that English stop words will be used, notably the ones imported at the beginning of the Python file, and script lines will be tokenized (only for custom sentiment score calculation purposes).

```
# Data Preprocessing
def do_data_preprocessing(df_script, stop_words):
    # convert to lower case
    df_script['Clean_Dialogue'] = df_script['Dialogue'].str.lower()
    # Remove punctuations
    df_script['Clean_Dialogue'] = df_script['Clean_Dialogue'].str.replace('[^\w\s]', ' ',
regex=True)
    # Remove spaces in between words
    df_script['Clean_Dialogue'] = df_script['Clean_Dialogue'].str.replace(' +', ' ',
regex=True)
    # Remove Numbers
    df_script['Clean_Dialogue'] = df_script['Clean_Dialogue'].str.replace('\d+', '',
regex=True)
    # Remove trailing spaces
    df_script['Clean_Dialogue'] = df_script['Clean_Dialogue'].str.strip()
    # Remove URLs
    df_script['Clean_Dialogue'] = df_script['Clean_Dialogue'].apply(lambda x:
re.split('https:\/\/.*', str(x))[0])
    # Remove stop words
    stop = stopwords.words('english')
    stop.extend(stop_words)
    df_script['Clean_Dialogue'] = df_script['Clean_Dialogue'].apply(
        lambda x: " ".join(x for x in x.split() if x not in stop))
    # Tokenize - Split into words
    df_script['Tokenized_Dialogue'] = df_script['Clean_Dialogue'].map(lambda x:
re.findall("[a-zA-Z0-9]+", x))
```

```
    return df_script;
```

The next function to be presented is optional and it is used for word frequency distribution. This function does not have any value within sentiment analysis, but it should exist as it might be used in feature enhancements/additions.

```
# Words Frequency Distribution (Optional)
def do_words_frequency_distribution(df_script):
    # join the words in string
    words = ' '.join(df_script['Clean_Dialogue'])
    words = words.split()
    # print(words)
    # create a empty dictionary
    data = dict()
    #  Get frequency for each words where word is the key and the count is the value
    for word in words:
        word = word.lower()
        data[word] = data.get(word, 0) + 1
    # Sort the dictionary in reverse order to print first the most used terms
    dict(sorted(data.items(), key=operator.itemgetter(1), reverse=True))

    return data;
```

In addition, the function that comes up next is optional and it is implemented for future use. Nevertheless, it produces a very nice figure presenting the word cloud of the script. This function depends on the previous one (do_word_frequency_distribution function) in order to be successfully executed. At this point, it should be pointed out that every output/outcome of the application is saved in the same location with the Python file.

```
# Wordcloud (Optional)
def do_word_cloud_presentation(data, movie_name):
    word_cloud = WordCloud(width=800, height=800, background_color='white',
max_words=1000)
    word_cloud.generate_from_frequencies(data)

    # plot the Word Cloud image
    fig = plt.figure(figsize=(10, 8), edgecolor='k')

    plt.imshow(word_cloud, interpolation='bilinear')
    plt.axis("off")
    plt.tight_layout(pad=0)

    fig.savefig('WordCloudPresentation' + '-' + movie_name + '.png')
    plt.close()
```

The following function lies at the heart of this implementation. In this function, the sentiment score calculation takes place. The results are saved in the original Pandas DataFrame that has been produced from script loading.

```
# Sentiment Analysis
def do_sentiment_analysis(df_script, positive_words, negative_words):
    df_script['positive'] = df_script['Tokenized_Dialogue'].map(lambda x: len([w for w in
x if w in positive_words]))
    df_script['negative'] = df_script['Tokenized_Dialogue'].map(lambda x: len([w for w
in x if w in negative_words]))

    # Method 1: Calculate absolute score by using positive count-negative count
    df_script['sentiment_score_1'] = df_script['positive'] - df_script['negative']
    df_script['SC_1'] = ['positive' if score > 0
                     else 'negative' if score < 0
                     else 'neutral'
```

```
                for score in df_script['sentiment_score_1']]

# Method 2: Normalise the score by the length of the line
df_script['sentiment_score_2'] = (df_script['positive'] - df_script['negative']) / (
   len(df_script['Tokenized_Dialogue']))
df_script['SC_2'] = ['positive' if score > 0
                else 'negative' if score < 0
                else 'neutral'
                for score in df_script['sentiment_score_2']]

# Method 3: Calculate the ratio of positive and negative words count
df_script['sentiment_score_3'] = df_script['positive'] / (df_script['negative'] + 1)
df_script['SC_3'] = ['positive' if score > 1.5
                else 'negative' if score < 0.5
                else 'neutral'
                for score in df_script['sentiment_score_3']]

# Method 4: Use TextBlob library
df_script['sentiment_score_4'] = [round(TextBlob(line).sentiment.polarity, 3)
                          for line in df_script['Clean_Dialogue']]
df_script['SC_4'] = ['positive' if score > 0
                else 'negative' if score < 0
                else 'neutral'
                for score in df_script['sentiment_score_4']]

return df_script;
```

A function that applies TF-IDF algorithm to the results of each sentiment score method conducted during a previous step was implemented. This function works as follows:

1. Split the dataset, which is the preprocessed dialogue along with the classification that the sentiment score method made (neutral, positive and negative)
2. Create a pipeline of two steps:
    a. Apply TF-IDF vectorizer

b. Apply Multinomial NB classifier
3. Train and test the model
4. Print and save the results

At this point, it is essential to note that all the parameters for the various methods used at this function are read from the configuration file.

```
# TF-IDF per single sentiment score method
def do_tf_idf(df_script, sentiment_column, sc_method, tf_idf_configuration,
tf_idf_model):
    print("\n<!> TF-IDF for Method: " + sc_method + " <!>")

    # split dataset
    X_train, X_test, y_train, y_test =
model_selection.train_test_split(df_script['Clean_Dialogue'],
                                                df_script[sentiment_column],
                                                train_size=tf_idf_configuration.train_size,
                                                test_size=tf_idf_configuration.test_size,

random_state=tf_idf_configuration.random_state)

    # create a pipeline of two steps:
    # 1. Apply tf-idf vectorizer to represent textual data in numeric vectors
    # 2. Apply MultinomialNB classifier, which is the scikit-learn implementation of
Multinomial Naive Bayes
    pipeline = Pipeline([
        ('vect', TfidfVectorizer(min_df=tf_idf_configuration.min_df,
                                 max_df=tf_idf_configuration.max_df,
                                 analyzer='word',
                                 lowercase=True,
                                 ngram_range=(
                                     tf_idf_configuration.ngram_range_start,
tf_idf_configuration.ngram_range_stop
```

```
                        ),
                        stop_words='english')),
    ('clf', MultinomialNB()),
])

# train the model
pipeline.fit(X_train, y_train)
feature_names = pipeline.named_steps['vect'].get_feature_names_out()

# test the model
y_predicted = pipeline.predict(X_test)

# print confusion matrix
print("\nConfusion Matrix: ")
confusion_matrix = metrics.confusion_matrix(y_test, y_predicted)
print(confusion_matrix)

# print the classification report
print("\nClassification Report: ")
classification_report = metrics.classification_report(y_test, y_predicted)
print(classification_report)

# print number of features and some samples
number_of_features = len(feature_names)
print("Number of features: ", number_of_features)

# calculate and print the model testing metrics
accuracy = np.round_(accuracy_score(y_test, y_predicted) * 100, 2)
precision = np.round_(precision_score(y_test, y_predicted, average='weighted') *
100, 2)
recall = np.round_(recall_score(y_test, y_predicted, average='weighted') * 100, 2)
f1 = np.round_(f1_score(y_test, y_predicted, average='weighted') * 100, 2)

print("Accuracy: ", accuracy)
```

```
    print("Precision: ", precision)
    print("Recall: ", recall)
    print("F1: ", f1)

    # write to table
    tf_idf_model.metrics_table.add_row([sc_method, number_of_features, accuracy,
precision, recall, f1])
    tf_idf_model.confusion_matrix_dic[sc_method] = confusion_matrix
    tf_idf_model.classification_report_dic[sc_method] = classification_report
```

A method that applies the do_tf_idf method (the previous one) to a movie was created. This method is responsible for saving the results for a single movie in files. One file for the classification report and one file for the confusion matrix will be created per movie containing results for all sentiment score methods that were used. Furthermore, this method is responsible for saving the metrics per movie to a single txt file. This file contains TF-IDF metrics for all sentiment score methods applied to all movies.

```
# TF-IDF per single movie
def orchestrate_tf_idf(df_script, movie_name, tf_idf_configuration):
    tf_idf_model = TfIdfModel(
        PrettyTable(["Method", "Number of features", "Accuracy", "Precision", "Recall",
"F1"]), {}, {}
    )

    do_tf_idf(df_script, "SC_1", "SC_1", tf_idf_configuration, tf_idf_model)
    do_tf_idf(df_script, "SC_2", "SC_2", tf_idf_configuration, tf_idf_model)
    do_tf_idf(df_script, "SC_3", "SC_3", tf_idf_configuration, tf_idf_model)
    do_tf_idf(df_script, "SC_4", "SC_4", tf_idf_configuration, tf_idf_model)

    print("\n")
    print(tf_idf_model.metrics_table)
```

```
    with open(tf_idf_configuration.metrics_export, 'a') as w:
        w.write("Movie: " + movie_name)
        w.write("\n")
        w.write(str(tf_idf_model.metrics_table))
        w.write("\n")
        w.write("\n")

    with open('TF-IDF-' + movie_name + '-confusion-matrix.txt', 'w') as w:
        for key in tf_idf_model.confusion_matrix_dic:
            w.write("Method: " + key)
            w.write("\n")
            w.write(str(tf_idf_model.confusion_matrix_dic[key]))
            w.write("\n\n")

    with open('TF-IDF-' + movie_name + '-classification-report.txt', 'w') as w:
        for key in tf_idf_model.classification_report_dic:
            w.write("Method: " + key)
            w.write("\n")
            w.write(str(tf_idf_model.classification_report_dic[key]))
            w.write("\n")
```

After having calculated sentiment scores using the different methods that have been chosen, the results need to be presented. The presentation will be conducted by visualizing them both in figures and in an Excel file using the function that follows. For every single movie script, a unique figure will be produced, showing the results of every sentiment score calculation method. Moreover, a single Excel file will be produced, in which every movie script will have a separate sheet for its own results.

```
# Visualizations
def do_visualizations(df_script, movie_name, writer):
    total = float(len(df_script))
```

```
new = df_script[['SC_1', 'SC_2', 'SC_3', 'SC_4']].copy()
print_dataframe(new)

ax = sns.countplot(x="variable",
                   hue="value",
                   data=pd.melt(new),
                   palette={"negative": "#FE2020",
                            "neutral": "#68BFF5",
                            "positive": "#BADD07"})
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_height() / total)
    x = p.get_x() + p.get_width()
    y = p.get_height()
    ax.annotate(percentage, (x, y), ha='right')

ax.set(xlabel='Sentiment Categories Methods', ylabel='Count')
ax.figure.savefig('Visualizations' + '-' + movie_name + '.png')

plt.close(ax.figure)

new = df_script[['SC_1', 'SC_2', 'SC_3', 'SC_4']].apply(pd.Series.value_counts)
new['SC_1 %'] = 100 * new['SC_1'] / new['SC_1'].sum()
new['SC_2 %'] = 100 * new['SC_2'] / new['SC_2'].sum()
new['SC_3 %'] = 100 * new['SC_3'] / new['SC_3'].sum()
new['SC_4 %'] = 100 * new['SC_4'] / new['SC_4'].sum()
new = new.rename(columns={"SC_1": "SC_1 Count", "SC_2": "SC_2 Count",
"SC_3": "SC_3 Count", "SC_4": "SC_4 Count"})
print_dataframe(new)

new.to_excel(writer, movie_name)
```

The orchestration and the implementation of the previously presented methodology will be done through the following function. This function is responsible for reading the configuration file (read_configuration function), deleting txt files for TF-IDF metrics, if any, opening an Excel file (the one that contains the results of sentiment analysis) and executing the following methods for every movie configured/parameterized in the configuration file: load_movie_script, do_data_preprocessing, do_words_frequency_distribution, do_sentiment_analysis, do_visualizations and orchestrate_tf_idf. Finally, this method saves the Excel file. As it might be observed, several "print" functions (print_dataframe and print_data functions) are used between others. These functions help to debug the application.

```
# Main
def main():
    nltk.download('stopwords')

    # Step #1: read configuration
    config_parser = read_configuration()

    basic_configuration = BasicConfiguration(
        config_parser.get(BASIC_CONFIGURATION_SECTION, 'stop_words'),
        config_parser.get(BASIC_CONFIGURATION_SECTION, 'positive_words'),
        config_parser.get(BASIC_CONFIGURATION_SECTION, 'negative_words'),
        config_parser.get(BASIC_CONFIGURATION_SECTION,
'movies_list').split(","),
        config_parser.get(BASIC_CONFIGURATION_SECTION,
'visualizations_export')
    )

    tf_idf_configuration = TfIdfConfiguration(
        config_parser.get(TF_IDF_CONFIGURATION_SECTION, 'train_size'),
        config_parser.get(TF_IDF_CONFIGURATION_SECTION, 'test_size'),
        config_parser.get(TF_IDF_CONFIGURATION_SECTION, 'random_state'),
        config_parser.get(TF_IDF_CONFIGURATION_SECTION, 'min_df'),
        config_parser.get(TF_IDF_CONFIGURATION_SECTION, 'max_df'),
        config_parser.get(TF_IDF_CONFIGURATION_SECTION, 'ngram_range_start'),
```

```
        config_parser.get(TF_IDF_CONFIGURATION_SECTION, 'ngram_range_stop'),
        config_parser.get(TF_IDF_CONFIGURATION_SECTION, 'metrics_export')
    )

    # Delete file if exists
    if os.path.exists(tf_idf_configuration.metrics_export):
        os.remove(tf_idf_configuration.metrics_export)

    # Step #2: process movies
    writer = ExcelWriter(basic_configuration.visualizations_export)

    for movie in basic_configuration.movies_list:
        movie_name = config_parser.get(movie, 'movie_name')
        script_path = config_parser.get(movie, 'script_path')

        print("\n--- Start working for movie " + movie_name + " ------------------------------
----------------------")

        # Step #3: load movie script
        df_script = load_movie_script(script_path)
        print_dataframe(df_script)

        # Step #4: data preprocessing
        df_script = do_data_preprocessing(df_script, basic_configuration.stop_words)
        print_dataframe(df_script)

        # Step #5: words frequency distribution (optional)
        data = do_words_frequency_distribution(df_script)
        print_data(data)

        # Step #6: word cloud presentation (optional)
        do_word_cloud_presentation(data, movie_name)

        # Step #7: sentiment analysis
```

```
        df_script = do_sentiment_analysis(
            df_script, basic_configuration.positive_words,
basic_configuration.negative_words)
        print_dataframe(df_script)

        # Step #8: visualizations
        do_visualizations(df_script, movie_name, writer)

        # Step #9: tf-idf
        orchestrate_tf_idf(df_script, movie_name, tf_idf_configuration)

        print("\n--- End working for movie " + movie_name + "---------------------------------
------------------------")

    writer.save()
```

The following IF statement is a common practice in order to run the code directly when using a Python file.

```
if __name__ == "__main__":
    main()
```

6.2 Sample Execution

The very first step that needs to be taken in order to execute this implementation is to install Python 3.9.2. Subsequently, the following commands have to be executed in order to install necessary Python libraries.

```
pip install openpyxl

pip install --user -U nltk

pip install pd
```

```
pip install wheel

pip install pandas

pip install TextBlob

pip install WordCloud

pip install seaborn

pip install sklearn

pip install numpy

pip install prettytable

pip install matplotlib

pip install scikit-learn
```

Prior to executing Application.py Python file, a configuration file named as MyApplicationProperties.txt, which exists in the same place, needs to be created. This configuration should have the following format:

```
[basic-config]
stop_words = < user-defined comma separated values >
positive_words = < user-defined comma separated values >
negative_words = < user-defined comma separated values >
movies_list=< user-defined comma separated values >
visualizations_export=< user-defined value of an Excel file >
[tf-idf]
train_size=< user-defined value >
test_size=< user-defined value >
random_state=< user-defined value >
min_df=< user-defined value >
max_df=< user-defined value >
ngram_range_start=< user-defined value >
```

```
ngram_range_stop=< user-defined value >
metrics_export=< user-defined value for TXT file >
[< user-defined movie #1 >]
movie_name = < user-defined movie #1 name >
script_path = < user-defined movie #1 script path >
[< user-defined movie #2 >]
movie_name = < user-defined movie #2 name >
script_path = < user-defined movie #2 script path >
```

Any text that appears between angle brackets and is highlighted in yellow color is user-defined and should be replaced by the user. A configuration file, having a format like the above, can have one or more movies configured in section blocks (Python language definition). One or more of these movies should be set in the variable movies_list in order to conduct sentiment analysis.

After all these have been set up, Application.py Python file can be run using the following command:

```
python Application.py
```

Upon successful execution of the application, no errors should appear in the command line. Furthermore, the following files will be created during the execution of the Application.py Python file:

1. One Excel file named after the value of visualizations_export variable that exists in the properties file. This Excel file will have as many sheets as the comma-separated values of the variable movies_list that exists in the properties file. Each single sheet will present results for a specific movie.
2. Many PNG files with prefix "Visualizations-" and the name of each movie set in variable movies_list exist in the properties file.
3. Many PNG files with prefix "WordCloudPresentation-" and the name of each movie set in variable movies_list exist in the properties file.

4. One txt file named after the value of metrics_export variable that exists in the properties file. This file contains TF-IDF metrics for all sentiment score methods applied to all configured movies.
5. Many txt files with the prefix "TF-IDF-", following the name of the movie and having the suffix "-classification-report" for each movie set in variable movies_list, exist in the properties file.
6. Many txt files with the prefix "TF-IDF-", following the name of the movie and having the suffix "-confusion-matrix" for each movie set in variable movies_list, exist in the properties file.

Bibliography

Ahmad, Munir & Aftab, Shabib & Muhammad, Syed & Awan, Sarfraz. (2017). *Machine Learning Techniques for Sentiment Analysis: A Review. INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING. 8. 2045-7057.*

Anikina, T. (2017, June). *Sentiment and Emotion Movie Script Annotation.*

Askitis, T. (2017, 03 05). *What is omophobia*. Retrieved from Medical Health of Sexual Health: https://www.askitis.gr/protothema/view/%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CE%B7-%CE%BF%CE%BC%CE%BF%CF%86%CE%BF%CE%B2%CE%AF%CE%B1

Chiny , M., Chihab, M., Chihab, Y., & Bencharef, O. (2021). LSTM, VADER and TF-IDF based Hybrid Sentiment Analysis Model. *(IJACSA) International Journal of Advanced Computer Science and Applications*.

Classic films, which today would have been condemned by the public, for messages of racism, sexism and pedophilia. (n.d.). Retrieved from Time Machine: https://www.mixanitouxronou.gr/klasikes-tainies-poy-simera-tha-eichan-katadikastei-apo-to-koino-gia-minymata-ratsismoy-sexismoy-kai-paiderastias/

Collomb, A., Costea, C., Joyeux, D., Hasan, O., & Brunie, L. (n.d.). Study and Comparison of Sentiment Analysis Methods for Reputation Evaluation.

Educational Theatre Association. (n.d.). *Historically banned and challenged plays*. Retrieved from Educational Theatre Association: https://www.schooltheatre.org/advocacy/local/freedomexpression/recentlybannedandchallengedplays

Frangidis, P., Georgiou, K., & Papadopoulos, S. (2020, May 6). Sentiment Analysis on Movie Scripts and Reviews. *Artificial Intelligence Applications and Innovations*, pp. 430-438.

Garcıa, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16(1), 1437-1480.

Hye-Yeon, Y., Moon-Hyun, K., & Bae, B.-C. (2017, December). Emotion and Sentiment Analysis from a Film Script: A Case Study. *Journal of Digital Contents Society*, pp. 1537-1542.

iTech. (2021, 05 23). *Which of The 3 Algorithms Models Should You Choose for Sentiment Analysis?* Retrieved from iTech: https://itechindia.co/blog/which-of-the-3-algorithms-models-should-you-choose-for-sentiment-analysis-2/

Kan, D. (2012). *Rule-based approach to sentiment analysis at ROMIP 2011.* DIALOGUE 2021.

Kim, H., & Jeong, Y. S. (2019). Sentiment Classification Using Convolutional Neural Networks. *applied sciences*.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining.* Morgan & Claypool Publishers.

Neethu & Rajasree. (2013). *Sentiment analysis in twitter using machine learning techniques. In 2013 fourth international conference on computing, communications and networking technologies (ICCCNT).*

Nhan Cach Dang, María N. Moreno-García, Fernando De la Prieta. (2020, Jan 31). Sentiment Analysis Based on Deep Learning: A Comparative Study. *Electronics*. Retrieved from https://bisite.usal.es/archivos/electronics-09-00483.pdf

Noble Desktop. (Jan28, 2. R.-W. (2022, Jan 28). *Noble Desktop. (Jan28,2022). Real-World Applications of Sentiment Analysis.* Retrieved from https://www.nobledesktop.com/classes-near-me/blog/sentiment-analysis-real-world-applications: https://www.nobledesktop.com/classes-near-me/blog/sentiment-analysis-real-world-applications

pandas development team. (n.d.). *pandas.DataFrame*. Retrieved from pandas: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

Sharma, D., Sabharwal, M., Goyal, V., & Vij, M. (2020, January). Sentiment Analysis Techniques for Social Media Data: A Review.

Sharma, M. (2020, May 10). *Sentiment Analysis: An Introduction to Naive Bayes Algorithm*. Retrieved from towards data science: https://towardsdatascience.com/sentiment-analysis-introduction-to-naive-bayes-algorithm-96831d77ac91

The Hellenic Parliament Foundation. (2018, Μαρψη 21). *"Aspects of racism" Educational Material and Laboratories of the Hellenic Parliamentary Foundation.* Retrieved from Social Policy.gr: https://socialpolicy.gr/2018/03/%CF%8C%CF%88%CE%B5%CE%B9%CF%82-%CF%84%CE%BF%CF%85-

%CF%81%CE%B1%CF%84%CF%83%CE%B9%CF%83%CE%BC%CE%BF%CF%8D-%CE%B5%CE%BA%CF%80%CE%B1%CE%B9%CE%B4%CE%B5%CF%85%CF%84%CE%B9%CE%BA%CF%8C-%CF%85%CE%BB.html

Towbin, M. A., Haddock, S. A., Zimmerman, T. S., Lund, L. K., & Tanner, L. R. (2003). Images of Gender, Race, Age, and Sexual Orientation in Disney Feature-Length Animated Films. *Journal of Feminist Family Therapy*.

Wikipedia. (2021, August 10). *Homophobia*. Retrieved from Wikipedia: https://el.wikipedia.org/wiki/%CE%9F%CE%BC%CE%BF%CF%86%CE%BF%CE%B2%CE%AF%CE%B1

Wikipedia. (2021, June 1). *Racism*. Retrieved from Wikipedia: https://el.wikipedia.org/wiki/%CE%A1%CE%B1%CF%84%CF%83%CE%B9%CF%83%CE%BC%CF%8C%CF%82