05/11/2019

Opinion Mining

Review Explorer

UNIVERSITY OF NICOSIA

Final year project submitted in partial fulfilment of the requirements of the degree of Bachelor of Science in Computer Science

Department of Computer Science
School of Sciences and Engineering
University of Nicosia May 2019

By

Grigorios Kokozidis

Project advisor:

______________________________	____________________	____ / ____ / ____
(Name)	(Signature)	(Date)

Examiner:

______________________________	____________________	____ / ____ / ____
(Name)	(Signature)	(Date)

Examiner:

______________________________	____________________	____ / ____ / ____
(Name)	(Signature)	(Date)

Department of Computer Science

School of Sciences and Engineering

University of Nicosia

Abstract

This thesis describes the Opinion Mining and sentiment analysis work on people`s opinions. A human being in all over the world express some opinion about what they like, or they do not like. In that opinion a person shows his/her feeling about that specific thing he/she is describing. That thing could be an item, a product, food, a service or whatever that is able to get own by a person. People express their feelings in different way from each other, that's why we all different. So, that's the reason why Natural Language Processing (NLP) is not an easy chapter at all. In this project we are going to explain and use some tools that are going to help us create a program that read and understand people`s opinions.

Contents

List of Figures, tables and Interfaces

Tables

Introduction

Chapter_1

1.0 The task of opinion mining has attracted interest during the last years. This is mainly due to the vast availability and value of opinions on-line and the easy access of data through conventional or intelligent crawlers. To utilize this information, algorithms make extensive use of word sets with known polarity. This approach is known as dictionary-based sentiment analysis. Such dictionaries are available for the English language. Unfortunately, this is not the case for other languages with smaller user bases. Moreover, such generic dictionaries are not suitable for specific domains. Domain-specific dictionaries are crucial for domain-specific sentiment analysis tasks. In this project we alleviate the above issues by proposing an approach for domain-specific dictionary building. We evaluate our approach on a sentiment analysis task. Experiments on user reviews on digital devices demonstrate the utility of the proposed approach.

Based on this idea and by using Domain-specific dictionaries this project is about Review Exploration. The key in this project is the connection of opinion words and opinions of users. Review explorer use opinion mining to illustrate its approach which is to help users on any environment to search fast and accurate about a specific item they need. This project has to do with human opinions which means is must be able to understand any logic behind every text-review. People has a different way to express a feeling or an opinion for anything is this world. Because of using the English language that almost is used in all over the world there will be a lot of different opinions on the internet. To understand every opinion, we learned and used sentiment analysis.

1.1 *Sentiment analysis*

Challenging task that attracted increasing interest during the last years. The availability of online data along with the business interest to keep up with consumer feedback generates a constant demand for online analysis of user-generated content. A key role to this task plays the utilization of domain-specific lexicons of opinion words that enables algorithms to classify short snippets of text into sentiment classes (positive, negative). This process is known as dictionary-based sentiment analysis. The related work tends to solve this lexicon identification problem by either exploiting a corpus and a thesaurus or by manually defining a set of patterns that will extract opinion words. In this work, we propose an unsupervised approach for discovering patterns that will extract domain-specific dictionary. The outcome is compared against lexicons extracted by the state-of-the-art approaches on a sentiment analysis task. Experiments on user reviews coming from a diverse set of products demonstrate the utility of the proposed method. An implementation of the proposed approach in an easy to use application for extracting opinion words from any domain and evaluate their quality is also presented.

Sentiment analysis is the process of extracting valuable information from user-generated opinionated content. This content may exist in various online sources and formats like product reviews, discussion forums or social networks. As the amount of online content grows, the importance of tools that automatically analyse such information sources become a necessity. Sentiment analysis lies in this category of tools, and it has high commercial value.

In this area many approaches have been developed. The dictionary-based approach is a representative example. In this approach, there is an opinion word lexicon which consists of two opinion word classes. Positive words such as “good”, “handy”, “impressive” follows by negative like “bad”, “useless”, “trash”.

Such generic lexicons are already available but, as expected, fall short in adapting to data coming from diverse domains. This requires the utilization of domain-specific dictionaries of opinion words

Currently, there are three main lines of research for tackling the task of opinion word extraction: (a) semantic thesaurus-based, (b) corpus-based and (c) pattern-based methods. These techniques although successful in some cases, they lack in some critical points. The first group relies on an information-rich semantic thesaurus that is not available for every language and domain. The disadvantage of the second group is that it captures only shallow dependencies. Finally, the third group (pattern-based methods) is using mostly predefined patterns that have limited applicability. Moreover, in most cases, these patterns are used solely for extracting opinion words and are not utilized in sentiment analysis.

Alleviating the disadvantages of the above methods, the main tool that our program uses (Didaxto) dynamically explores the pattern space by analysing the corpus in an unsupervised manner. Moreover, the extracted patterns have the following disadvantages: (a) they are corpus and domain specific and (b) they can be utilized directly in sentiment analysis.

Review Explorer is about to read any text and understand if that text has positive, negative or even neutral behaviour. The main idea is to cluster opinions of users that are talking to a specific target. Is a program that does more than to read and connect opinions. There are more functionalities built in it that make it a program within opinion mining space. Review Explorer use "Didaxto", a program that utilizes opinion modifies, sentiment consistency theories, polarity assignment graphs and pattern similarity metrics. We use the power of Didaxto to make some very good and efficient dictionaries that will hold opinion words in specific categories. The actual output of our program is the grouped texts that we will have by just holding the proper opinion words. Some words may have a positive meaning in a text and in other it could have a negative, like the word "low".

This word “low” in a positive text can be expressing the low bad properties that an item has for example and in a negative text it could be the low good properties. So, in situation like this we cannot be expecting our program to understand a text only by one word but also by more.

1.2 ***Didaxto***

Didaxto is a software that implements and integrates the Sentiment analysis task in an easy to use graphical user interface. Provides an unsupervised, multistage, resource efficient pattern extraction method. The extracted patterns can be utilized in extracting domain-specific opinion words and in sentiment classification. Our comparative study demonstrates that our approach outperforms two state-of-the-art methods in extracting opinion words and sentiment analysis. Also, it provides a dataset of opinions in the English language. The dataset can serve as a benchmark in a variety of opinion mining tasks**.**

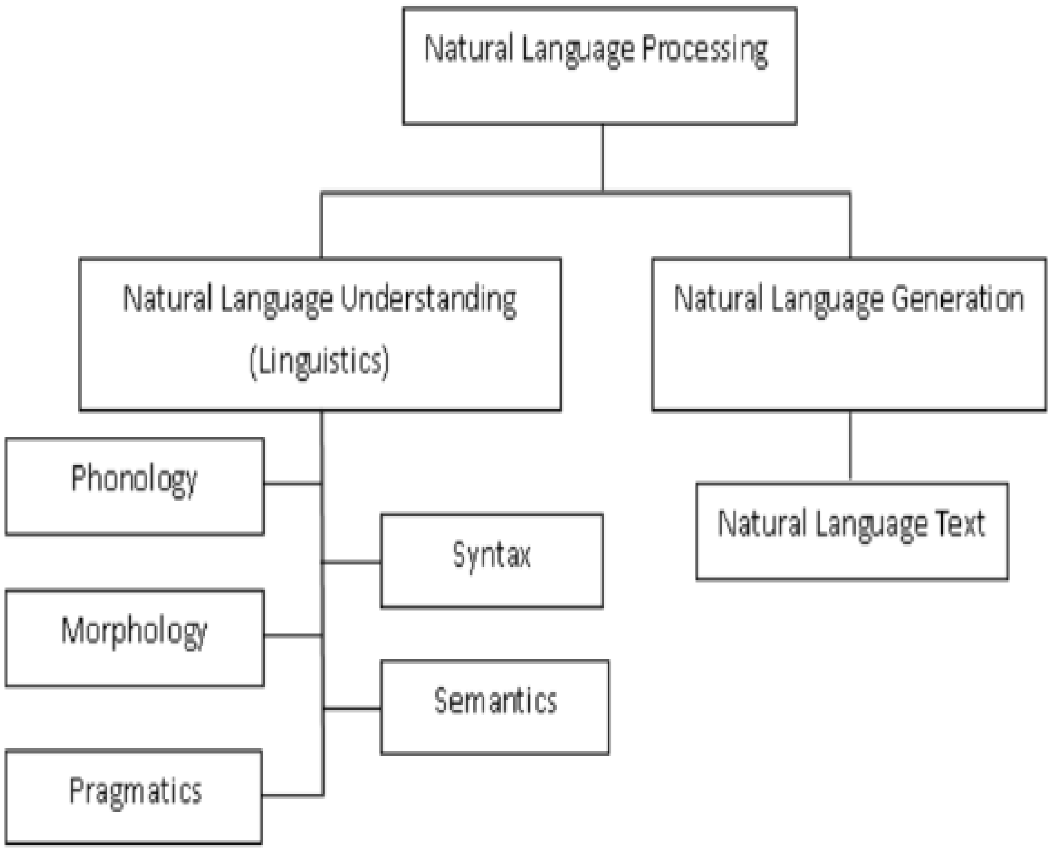
There are many services that are already can do this task nowadays. Software developers with linguistics developed a mechanism to make the computers to understand the human language, they called it natural language processing. NLP is very important chapter of Artificial Intelligence and machine learning. To train a computer or a machine to understand the human language is not an easy task at all. Every human being on this planet has his/her own way of expression. A person who lives in a different country or in a different city or in the same city with other person but in different neighbourhood they could also have different way of speaking with each other. So that means to train a single machine to understand all those people`s different way of speaking is a such a hard task. However, you cannot create a different machine for each country because this would be much harder and expensive. The goal is to create one machine so smart that could be understand any human writing style.

How developers can achieve this goal? We all know that the English language is the international language, so the first target is to train a computer to read and understand the English language. Developers are pretending as the teachers of computers, that means a computer will learn to read English language if a developer understand it very good as well. A programmer should not go again to school and study grammar and vocabulary, but he/she can cooperate with a Professional Linguistic as a team. Although there are so many papers and documents out there that will help any programmer to make a research and create the proper algorithms. NLP and Google Cloud Platform are some services that solved this problem for software developers.

1.3 ***Natural Language Processing***

Natural Language Processing (NLP) is a tract of Artificial Intelligence and linguistics, devoted to make computers understand the statements or words written in human languages. Natural Language Processing came into existence to ease the user`s work and to satisfy the wish to communicated with the computer in natural language. Since all the users may not be well-versed in machine specific language, NLP caters those users who do not have enough time to learn new languages or get perfection in it.

A language can be defined as a set of rules or a set of symbols. Symbol are combined and used for conveying information or broadcasting the information. Symbols are tyrannized by the Rules. Natural Language Processing basically can be classified into two parts i.e. Natural Language Understanding and Natural Language Generation which evolves the task to understand and generate the text.


Natural Language Processing
Natural Language Understanding (Linguistics)
Natural Language Generation
Phonology
Syntax
Morphology
Semantics
Pragmatics
Natural Language Text

Natural Language Processing

Figure 1.0

1.4 ***Linguistics***

Science of language which includes Phonology that refers to sound, Morphology word formation, Syntax sentence structure, Semantics syntax and Pragmatics which refers to understanding.

“Noah Chomsky”, one of the first linguists of twelfth century that started syntactic theories, marked a unique position in the field of theoretical linguistics because he revolutionised the area of syntax. Which can be broadly categorized into two levels Higher Level which include speech recognition and Lower Level which corresponds to natural language. Few of the researched tasks of NLP are Automatic Summarization, Co-Reference Resolution, Discourse Analysis, Machine Translation, Morphological Segmentation, Named Entity Recognition, Optical Character Recognition, Part of Speech Tagging etc. Some of these tasks have direct real-world applications such as Machine Translation, Named entity recognition, Optical character recognition etc. Automatic summarization produces an understandable summary of a set of text that provides summaries or detailed information of text of a known type. Co-reference resolution refers to a sentence or larger set of text that determines which word refer to the same object. Discourse analysis refers to the task of identifying the discourse structure of connected text. Machine translation which refers to automatic translation of text from on human language to another. Morphological segmentation which refers to separate word into individual morphemes and identify the class of the morphemes. Named entity recognition (NER) it describes a stream of text, determine which items in the text relates to proper names. Optical character recognition (OCR) it gives an image representing printed text, which help in determining the corresponding or related text. Part of speech tagging it describes a sentence, determines the part of speech for each word. Though NLP tasks are obviously very closely interweaved but they are used frequently, for convenience. Some of the tasks such as automatic summarisation, co-reference analysis etc. act as subtask that are used in solving larger tasks.

The goal of Natural Language Processing is to accommodate one or more specialities of an algorithm or system. The metric of NLP assess on an algorithmic system allows for the integration of language understanding and language generation. It is even used in multilingual event detection, purposed a novel modular system for cross-lingual event extraction for English, Dutch and Italian texts by using different pipelines for different languages. The system incorporates a modular set of foremost multilingual Natural Language Processing (NLP) tools. The pipeline integrates modules for basic NLP processing as well as more advanced tasks such as cross-lingual named entity linking, semantic role labelling and time normalization. Thus, the cross-lingual framework allows for the interpretation of events, participants, locations and time, as well as the relations between them. Output of these individual pipelines is intended to be used as input for a system that obtains event centric knowledge graphs. All modules behave like UNIX pipes: They all take standard input, to do some annotation, and produce standard output which in turn is the input for the next module pipelines are built as a data centric architecture so that modules can be adapted and replaced. Furthermore, modular architecture allows for different configurations and for dynamic distribution.

Most of the work in Natural Language Processing is conducted by computer scientists while various other professionals have also shown interest suck as linguistics, psychologist and philosophers etc. One of the most ironical aspect of NLP is that it adds up the knowledge of human language. The field of Natural Language Processing is related with different theories and techniques that deal with problem of natural language of communicating with the computers. Ambiguity is one of the major problems of natural language which is usually faced in syntactic level which has subtask as lexical and morphology which are concerned with the study of words and word formation.

Each of these levels can produce ambiguities that can be solved by the knowledge of the complete sentence. The ambiguity can be solved by various methods such as Minimising Ambiguity, Preserving Ambiguity, Interactive Disambiguated and Weighting Ambiguity. Some of the methods proposed by researchers to remove ambiguity is preserving ambiguity, their objectives are closely in line with the last of these: they cover a wide range of ambiguities and there is a statistical element implicit in their approach.

Opinion Mining

Chapter_2

Abstract

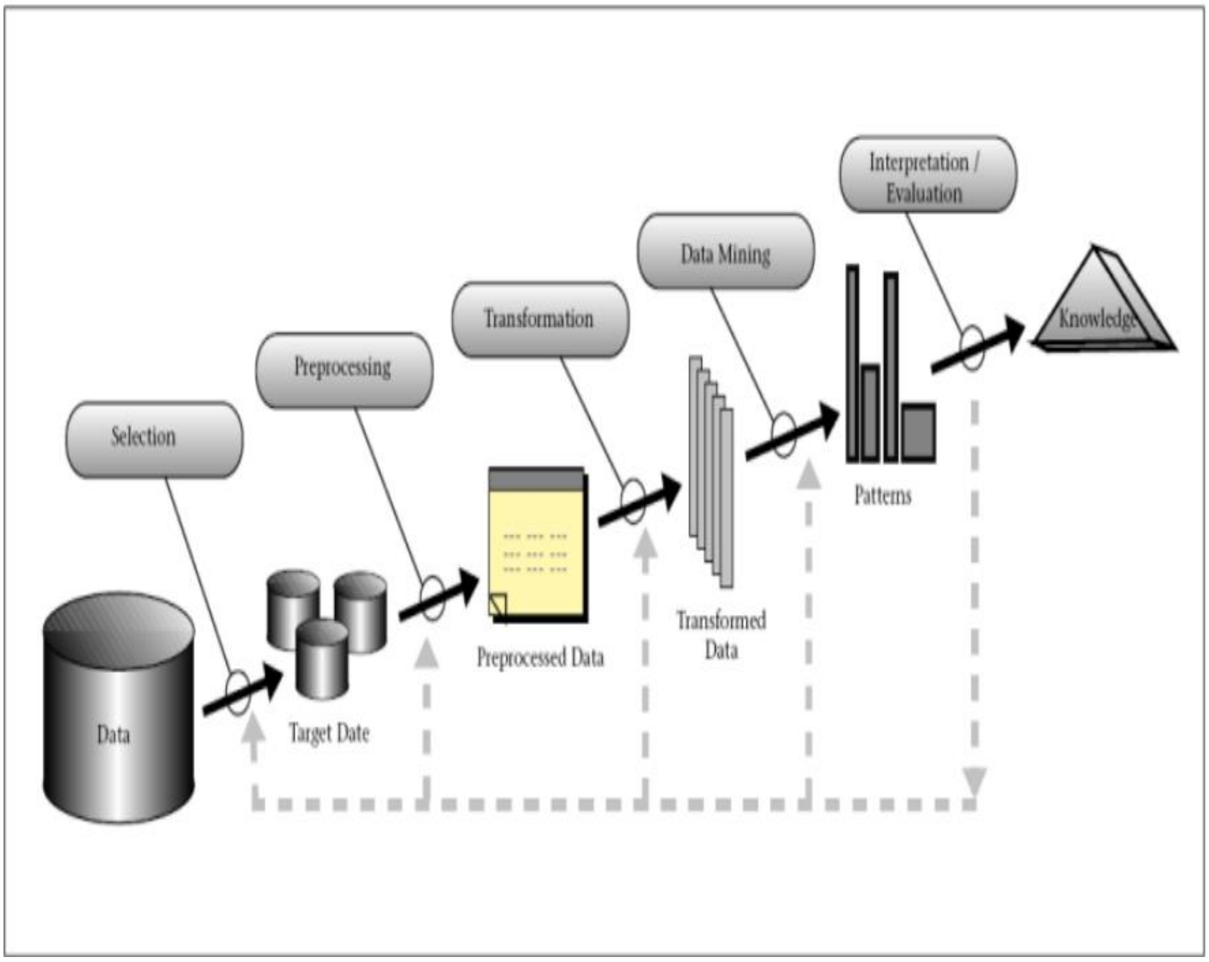
2.0 Opinion mining is critical for both individuals and companies. Individuals want to analyse other`s opinions about a product or a service that they want to buy, so they can decide if they will buy it or not. On the other hand, companies want to analyse the feedback that they get from their customers about their products or services to make future decisions. So, analysing customer`s opinion and his/her response is an important task. Reviews are available on different web forums, blogs and product review sites. Mining is used on product reviews to evaluate opinions of customers. This approach helps customers to read reviews of other customers that already got a specific product or service to understand if it's worth it to buy or not. Companies uses this approach to make their product`s features more satisfying for their customers. "This leads to overcome the requirements of marketing intelligence and product benchmarking in the production industry". In this project we do a small survey of how opinion mining is an extremely challenging task and how we extract opinion words and opinion targets from people`s reviews.

2.1 ***What is Opinion mining***

Involves Text Mining and Language Processing (NLP) and Text Classification. Text Mining has a great potential to overcome the current deficiencies. Unfortunately, natural language processing (NLP) encounters a range of difficulties due to the “sophisticated” nature of human language. Moreover, the area of opinion mining involves the problem of Text Classification, which is totally different from the usual Text Mining. In usual text classification, the focus is on identifying topic, whereas for opinion mining, Sentiment Classification is one which focuses on the assessing writer`s sentiment toward the topic. Emotions are not satisfactorily analysed with keyword-based methods.

In opinion mining we use some powerful techniques to gather and understand data information. “Knowledge Discovery” is the extraction of hidden, previously unknown information from data that is of interest to the end user`s knowledge and objectives. In modern database technology, processing of very huge volume of data is involved, to extract new knowledge.

As knowledge is not always represented explicitly, so to extract useful information from complex and large volume of data, current database technology involves processing of data. So, Data mining and knowledge discovery (KDD) process is used for this purpose and the field of data mining and knowledge discovery from databases has emerged as a new discipline in engineering.



Overview of the steps consulting the KDD process.

Figure 2.0

We learned about KDD process in Data mining course at university and steps that are shown above in the figure. KDD process is a fundamental knowledge that is important to analyse data and extract critical information from it.

The KDD process includes the following steps:

1. **Learning the application domain:** includes relevant prior knowledge and the goals of the application.
2. **Creating a target dataset:** includes selecting a dataset or focusing on a subset of variables or data samples on which discovery is to be performed.
3. **Data cleaning and pre-processing:** includes basic operations such as removing noise or outliers if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handing missing data fields, and accounting for time sequence information and known changes, as well as deciding DBMS (Database management systems) issues such as data types, schema, and mapping of missing and unknown values.
4. **Data reduction and projection:** include finding useful features to represent the data, depending on the goal of the task, and using dimensionality reduction or transformation methods to reduce the effective number or variables under consideration or to find invariant representations for the data.
5. **Choosing the function of data mining:** includes deciding the purpose of the model derived by the data mining algorithm e.g. this can be summarization, classification, regression, and clustering.
6. **Choosing the data mining algorithm(s):** includes selecting method(s) to be used for searching for patterns in the data, such as deciding which models and parameters may be appropriate (e.g., models for categorical data are different from models on vectors over real) and matching a particular data mining method with the overall criteria of the KDD process (e.g., the user may be more interested in understanding the model than in its predictive capabilities).
7. **Data mining:** includes searching for patterns of interest in a representational form or a set of such representations, including classification rules or trees, regression, clustering, sequence modelling, dependency, and line analysis.

8. **Interpretation:** includes interpreting the discovered patterns and possibly returning to any of the previous steps, as well as possible visualization of the extracted patterns, removing redundant or irrelevant patterns, and translating the useful ones into terms understandable by users.

9. **Using discovered knowledge:** includes incorporating this knowledge into the performance system, taking actions based on the knowledge, or simply documenting it and reporting it to interested parties, as well as checking for and resolving potential conflicts with previously believed (or extracted) knowledge.

2.2 ***Problem Description:***

First, before to start thinking of build a project that has to do with the topics that we described above you have to learn them and understand clearly what their approaches are. One of the most basic knowledge that a programmer must know before to start working on this field is to has at least some basic knowledge of linguistics, that would be ideal. Any programmer though prefers to find the information and apply it to his/her program. To achieve the best Opinion Mining project programmers and linguistics must be in the same team.

Review Exploration is using tools of opinion mining and natural language processing. As we described above the natural human language is not an easy task at all. To make a virtual machine or a mechanism that is going to apply to a software or a service (API) is something that needs to take a lot of things under consideration. Our project, Review Exploration, has multiple targets and goals to achieve.

So, how review exploration interacts with opinion mining, and why we need opinion mining to achieve our goals?

Most important goal of our project is to create a user-friendly environment where an individual user can extract reviews from a specific location (url or database) and analyse them fast and accurately. When we say fast, we mean the good performance of our software. To have good performance we must use good technologies and tools that will help our program run smoothly on any system. The functionality must be built and separated in small pieces that will be efficient and clear. Our program must be as accurate as it possible, meaning that we must give back to the user the result that he/she is searching for. Accuracy of output will be achieved by understanding the workflow of project and design the structure properly. Studying opinion mining and data mining KDD processing helped us a lot of how we going to design the workflow of our project.

Review exploration must be smart software that is going to do some human functionalities. The basic functionality that we must to implement is to train our program to read any English text. As we were kids we learned to read a simple and easy texts. Our teachers taught us first the alphabet, after some words and later we could read whole sentences. So, the approach is similar in our problem as well. If it was difficult for us as a human being to read whole sentences from the beginning imagine how difficult or let me say unfortunately for a software to do that. So, we need to find some dictionaries with different categories of words (adverbs, verbs, comparatives, positives, negatives etc) and teach our machine how to connect them in one sentence and what is their meaning. Huge problems are difficult to solve because they are huge, so we must break our problem in small pieces and start to find solution for each one. We are going to talk about people`s reviews of any products or services. A review could be any size, some people express their thoughts by talking more and some other by talking less.

So, we are going to deal with any type of a text review that a customer/user gave as a feedback. Obviously small size of review it would much easier for our program to analyse and give the proper output we want, but there could be some small reviews with very difficult meaning. A customer/user/client can write anything he/she wants depends of what he/she had in his/her mind. That means that we could find a big size of review that is written in simple and easily understandable words instead of a smaller size of review that has only difficult words or the meaning of those words are something that our software cannot recognise easy. So, the size of a review matters only for the performance of the program, as bigger is the text as much time it will be taken to be read.

People write a text in the paper with their own writing style. The fact that comments or views entered by people who are different from each other in the way they write, their use of language, abbreviations and their knowledge is a challenge on its own. People also do not express opinion in the same way. One might use certain negative terms in a sentence Text that appears in an online newspaper and that which appears in a n online forum is widely different. The same could be done by writing a text from the keyboard. Some people may write a text in a formal structure but some other in informal structure. Informal text It could be a text without spacing, no capital letters in the beginning of a sentence, no commas and dots etc. In cases like this a smart program should has a mechanism that simplifies any text that is going to read it. By using a simple format for all reviews, we avoid any misunderstandings texts and trying to be as accurate as we can in the specific scope of the content. After we get the text in the format we need, we must make a study on some definitions that we have to know about Opinion Mining. As people are free to give their opinions on anything e.g., they buy a product and then they express their views on products features in various forums.

So, at this point we have the person that buy a product, which is the opinion holder, the opinion that the opinion holder is going to express about that product and the object that could be a product with some attributes. The term "object" is used for the entity on which comments have been given, "Opinion" which is thoughts of a person about something, "Opinion Holder" is the person giving his/her opinion on something is the holder of the opinion and the semantic orientation that is an opinion on a feature "x" states whether the opinion is positive, negative or neutral. This classification can be done at sentence level whether a sentence contains a positive opinion on a feature of an object, or it may contain negative opinion on it. An object "A" can be represented with a set of features, F={f1,f2…fn} which includes the object itself. Each feature f1 that belongs to F can be expressed with a finite set of words or phrases which are synonyms. That is, there is a set of corresponding synonym sets for n features. An opinion holder comments for each feature to describe the feature by choose any word

he/she thinks is suitable for his/her expressions, that can be positive, negative or neutral. Opinion mining is used to extract useful information from any opinions that will make our program to understand the orientation, target and the feeling of the person in his/her opinion. Also, sentiment analysis system is suggested for classifying products features in consumer reviews by means of mining class association rules.

We can understand as humans that an opinion is positive, negative or neutral from the reviewer`s speech. As humans if we hear for example words like "good", "amazing", "cool", "nice" we can understand that that person is talking positive about something. Else if a person says words like "bad", "terrible", "horrible", "slow" we can understand that he/she is talking negative about something. Neutral opinion is something in the middle like "it's good but it could be better" or "it's nice but it could be having more of that", because as we were kids, we get trained to understand the meaning of words and while we are growing, we learn more words.

The same happens with a program, instead of talking to a computer to train it English vocabulary you must implement some algorithms. Also create a database that contains the starter pack of words and by time passing you should update this database importing new words.

2.2.1 ***Mining comparative and superlative opinion sentences:***

A person who purchased a certain product can express direct opinion on the features of the object and give comments like “Feature1 of this product is great”, “Feature2 of this product is so so”, “Feature3 of this product is bogus”. Such opinions are direct or regular opinions. In this case, during opinion mining, objects, their features and the orientation of the opinion are to be identified. Such opinions are expressed by using third form of adjectives/adverbs e.g. “Feature4 is best”, “Feature5 is words”. Some people express their views on products in a comparative way e.g. Feature1 of product is “better” than Feature1 of another product. Thus, comparisons are made using comparison words and second form of adjectives/adverbs e.g. “better” in the last sentence. Comparisons ae related to but are also different from direct opinions. In such text it is to be identified that what objects are being compared, which of their featured are being compared and which objects are given preference by their opinion holders. If we take a camera as example in any online marketplace, we will find a lot of positive and negative opinions. A camera has as important features the “picture quality” and “size”. There will be a lot of reviews about a specific feature or for all the features of the camera. Any person that wants to buy that camera he/she can look on those reviews and by reading some of them he/she can understand if people that have already bought that camera are satisfied or not.

Another way of summary of opinions gathered from multiple reviews on a particular camera is shown in Table 1. These are regular opinions marking features of a product as good or bad. The object features and number of positive and negative reviews on these features are listed in the table below.

Product feature	**Reviews (positive)**	**Reviews (negative)**
Camera	125	7
Picture Quality	123	6
Size	82	10

Table 1. Number of Positive and negative reviews on camera features.

This summary can be easily visualized using a bar chart Figure 2 – 2.1 shows such a chart. In the figure, the extent of the bar above the X-axis show number of positive opinions on a feature (shown at the top of the bar), and the extent of bar below the X-axis shows the number of negative opinions on the same feature.

Obviously, other visualizations are also possible. For example, one may only show the percentage of positive (or negative) opinions on each feature. Comparing opinion summaries of a few competing objects is even more interesting. Figure 2.1 shows a visual comparison of consumer opinions o two competing digital cameras. One can clearly see how consumers view different features of each camera.

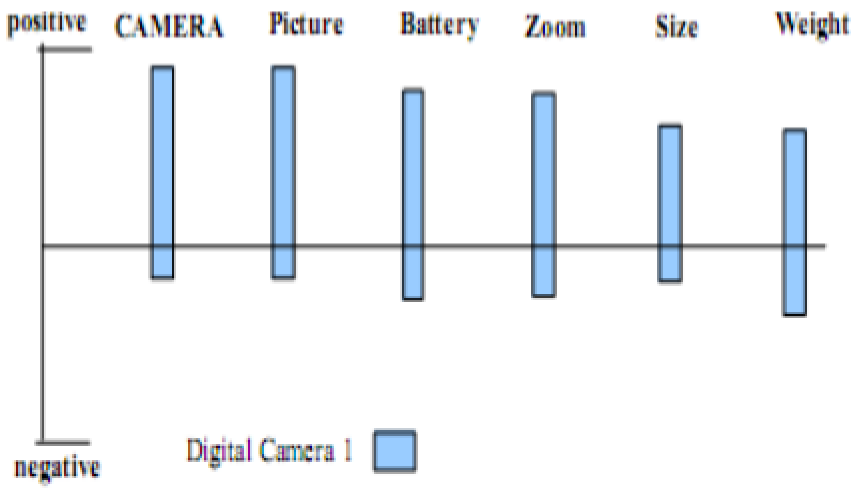


Figure 2.1 Visual Feature-based summary of opinions on a digital camera.

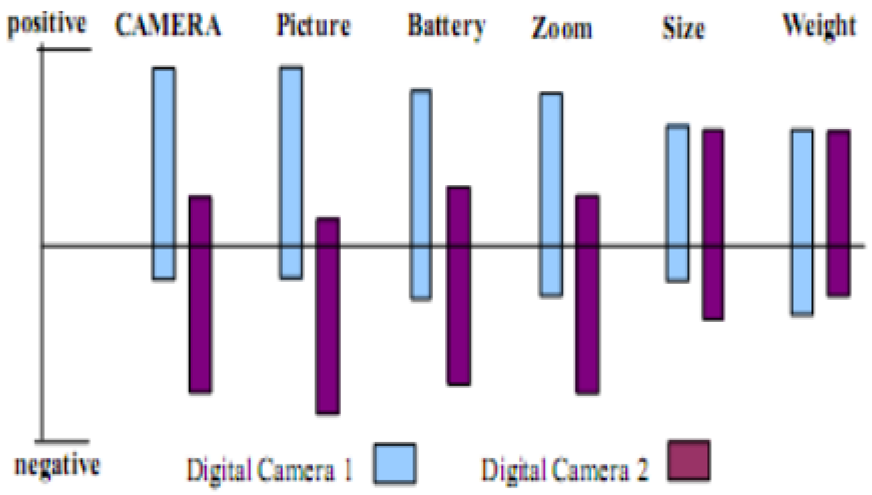


Figure 2.2 Visual Opinions Comparisons of two digital cameras.

2.2.2 ***Object identification:***

In opinion mining, firstly you must identify the objects in a review on which opinion have been given. This problem is important because without knowing the object on which an opinion has been expressed, the opinion is of little use. However, there is a difference, for opinion mining only those objects in the review are to be considered which are in competition to each other. The system thus needs to separate relevant objects and irrelevant objects.

2.2.3 ***Feature extraction:***

“The voice on my phone was not clear” this is an opinion that contains an object which is the “phone”, also contains a feature which is the “voice” and an opinion word which is the “clear”. Some techniques frequently used nouns noun phrases as features are identified as features, which are usually genuine features. In this case we must break the sentence and get the information that it has. The information we are looking for is at least one “Opinion word” and an “Opinion Target” which can be a feature of the product or the product itself. In this sentence we have a feature and the product, but our target is the product`s feature (voice). The opinion word “clear” is a positive word but its orientation is not positive. The person who expressed this opinion is talking about a bad feature on his/her mobile phone and we know that because he/she is using negation word “not” before opinion word. This makes us think about how to make our program to understand situations like this.

2.2.4 ***Orientation identification:***

Different words or phrases can be used to refer to the same feature of the object. So, much words (synonyms) should be identified and grouped together. It is a difficult task to identify these words. A lot of research is required to be done on this issue as it has not been much addressed in the past. It is needed to group synonym features, as people often use different words or phrases to describe the same feature. In the example above, “voice” and “sound” both refer to the same feature. Opinion orientation identifies the orientation of opinions, determines whether the opinions on the features are positive, negative or neutral. Many approaches can be used for this purpose. Usually, lexicon-based approach is used as it performs quite well. The lexicon-based approach basically uses opinion words and phrases in a sentence to determine the orientation of an opinion on a feature. A relaxation labelling based approach is also proposed. Classifying an opinion as positive, negative or neutral can be a difficult task in opinion mining. A word could be considered positive in one situation and negative in another situation. This can be difficult to calculate as a sentence can be considered negative because of the use of negative words in it.

Existing approaches are based on supervised and unsupervised methods. One of the key issues is to identify opinion words and phrases (e.g. good, bad, great, poor), which are instrumental to sentiment analysis. The problem is that there are seemly an unlimited number of expressions that people use to express opinions and in different domains they can be significantly different. Even in the same domain, same words depict different opinions depending upon the context of the text. For example, if a person a says “camera has a `small` size”. Then `small` has a positive orientation because for a camera to have small size is a positive feature as makes it easily portable. But if another person says, “Camera has a `small` battery like”.

Then `small` has a negative orientation because for a camera to have small battery life a negative feature as it needs to be replaced frequently. So, at this point it is extremely hard to create an algorithm that is going to determine for each object which are the positive features, and which are the bad features. There are millions of products out there with millions of features.

To integrate the above tasks is also complex because we need to have the basic information from a review or a text to make the ideal analyzation. Ideal review or text should include the opinion (1) that must be given by opinion holder (2) on a feature (3) of object (4) at time (5). To make matters worse, a sentence may not explicitly mention some pieces of information, but they are implied due to pronouns, language conventions, and the context. There is nowhere in any forum a warning description that alert people to write their opinion in a specific format, so people are free to express their opinion in any way they think is correct for them. To deal with these problems, we need t apply Natura Language Processing (NLP) techniques in the opinion mining context, e.g., parsing, word sense disambiguation, and coreference resolution. We use coreference resolution as an example to give a glimpse of the issues. For our example blog, to figure out what is “my phone” and what is “her phone” is not a simple task. Coreference resolution is a classic problem in NLP.

Using Opinion mining at a document level is not a good idea for an advance program. Classifying evaluative texts at the document level does not five information about likes and dislikes of the opinion holder.

An evaluative document usually contains both positive and negative opinions. In the document, Opinion holder does not have all the positive opinions on features of the objects, nor does he have disliking for all the features, although the general sentiment on the object may be positive or negative.

So, to deal with such a problem, opinion mining at feature level can help to identify both positive and negative opinions in a document. So, feature level approach solves this issue.

Opinions change with time. Another challenge lies in the issue of being able to monitor changing with the passage of time. This helps us to observe if a certain product gets improved with time, or people change their opinion about a product and get convinced for it with time. Some research work is done to identify how the people`s mood changes over time. This work done observes blogs where the mood is explicitly specified either by selecting from a predefined list of moods or by entering it as free text.

Another challenge that faced in opinion mining is about identification of the strength of an opinion. The strength of an opinion can change as the discussion continues in a forum i.e. arguments used during discussion are strong enough to change the strength of opinions. To identify strength of opinions there is an application called *SentiWordNet*. Our program in current version is not include services of this application but in future version it could be implemented as well.

A bigger challenge for opinion mining comes when people express positive and negative review in the same sentence. This is mostly the issue when people communicating through informal mediums like blogs and forums.

People are more likely to combine different opinions in the same sentences. Such sentences can be difficult to parse for opinion mining. Sentiment mining or opinion mining is contrasted generally with the traditional fact-based text mining. Text mining seeks to classify documents by topics domains and users. Strength of feeling, degree of positivity and similar factors can be of potential importance in opinion mining.

Misleading opinions due to sarcastic and ironic statements exists in text. In such scenario, positive words can be negative sense of usage in a metaphorical manner. So, text in statement can be hard to identify as sarcastic or ironic which can lead to erroneous orientation and misleading opinion mining.

2.3 ***Opinion Exploration***

People cannot take a decision easy about to buy something or not. They must be sure that the product they are going to buy is reliable and the services its providing are good. So, from this point we are going to explain more detailed about our project “Review Explorer”. Review Explorer is about connecting people`s opinions. Is like a guy, our friend or anybody, who always helps us when we need a secondary opinion about something. If you are searching about a specific issue on a product that guy is going to give you his/her thoughts or even better gets you in touch with other people that can help you with your problem. So basically, Review Explorer is a “guy” where we can search and find opinions that are going to help us to have a clear thought about a specific product.

Recommender systems are inescapable in a wide range of web applications, e.g. Amazon or Netflix, to provide users with books or movies that match their interest. Accurate recommendations generate returns of investments up to 30% due to increased sales. Many such systems rely on collaborative filtering approaches that recommend items based on user rating history. Concomitantly, the rising popularity of social networks has provided new opportunities to filter our relevant content for users.

That means people follow the recommendations and the good recommendations brings more incomes to the companies. Our project`s duty is to get one dataset of opinions – reviews, or more, and use some tools to analyse and get some results about the recommendations that people gives in those datasets. All datasets are medium to large-scale and exhibit various properties regarding user social ties and items rating. Datasets usually must include the path, from where this opinion is taken, opinion rate, opinion. Datasets of this format Is extracted from Didaxto which is a program that uses some tools to analyse reviews and get critical information from them. In Didaxto you can also create a dataset, as we mentioned above, graphs representing the positivity or negativity of an opinion and the most important for our project the opinion words from each review.

Didaxto

Chapter_3

Abstract

3.0 Didaxto implements an unsupervised approach for discovering patterns, that will extract domain-specific dictionary from product reviews. The approach utilized opinion modifiers, sentiment consistency theories, polarity assignment graphs and pattern similarity metrics. Apart from extracting the dictionary of opinion words, the application allows the evaluation of the dictionary by means of a sentiment classification task on product reviews.

3.1 ***Architecture***

Didaxto represents sentiment analysis in a graphical user interface. Sentiment analysis is the task of extracting valuable, non-trivial knowledge from a collection of documents containing opinions. Opinions can refer to a product, that could be an item or a service or even political figures. Advanced algorithms can discover opinions of multiple features for the entity under investigation. These techniques are applied to data extracted from various sources like discussions boards (forums), social networks (facebook, twitter), blogs or video sharing networks.

Sentiment analysis is utilized in different levels of granularity: at sentence-level, document-level or corpus level. A key role to all these levels plays the so-called list of "Opinion words". This is a dictionary containing terms of known polarity. In most cases, the list is dual. A positive wordlist ("great", "wonderful", "Nice") is coupled with a negative wordlist ("ugly", "bad", "slow"). Neutral opinions do not contain any "neutral opinion word" but they contain phrases that makes us understand that an opinion is neutral, so we do not have a wordlist of neutral words.

Sentiment analysis requires a high quality of “opinion word” dictionaries since algorithms depend their initialization, enhancement and operation on them. Such dictionaries are already available for the English language. Many of them are manually constructed.

However, the availability of opinion wordlists for less popular languages is very limited. If a software developer wants to apply any other language dictionaries in his/her project then he/she must do a huge research about their language opinion words.

As we discussed in previous chapters, a single word has its own meaning but when it appears in a sentence and there are other words before and after it then its meaning may be different. This is a very important issue such lists is that they are domain dependent. Therefore, approaches that can extract domain-specific opinion dictionaries are necessary.

Didaxto is a program that uses multi-stage iterative approach that gets a small seed of generic opinion words as input and extends it with domain-specific words. It utilizes language patterns and takes advantage of a double propagation procedure between opinion words and opinion targets. The effectiveness of the approach is estimated in a sentiment analysis task. Results justify the utility of all steps of the proposed algorithm. In addition, didaxto is a software that enables the use of the above techniques under a user-friendly interface.

In a probabilistic method is presented that builds an opinion word lexicon. The method uses a set of opinion documents which is used as a biased sample and a set of relevant documents as a pool of opinions. In order to assess the effectiveness of the algorithm a dictionary made up of 8K words is used.

Certain probabilistic functions such as Information Content, Opinion Entropy and Average Opinion Entropy are used as extraction tools. The method is based upon the observation that nouns contain high information value, while adjectives, adverbs and verbs provide additional information to the context.

Upon these observations and the probabilistic tools, they extract the opinion word lexicon.

The approach, of creators of the Didaxto Pantelis Agathangelou and my supervisor Dr.Ioannis Katakis, comprises of a series of steps that gradually detect opinion words. Each step creates a pool of opinion words that will constitute the feed for the next detection step. More specifically the construction of the proposed algorithm can be summarized in the following steps:

1. Opinion Pre-processing
2. Auxiliary List Preparation
3. Seed Import and Filtered Seed Extraction
4. Conjunction Based Extraction
5. Double Propagation
6. Opinion Word Validation

3.1.2 ***Opinion Pre-processing:***

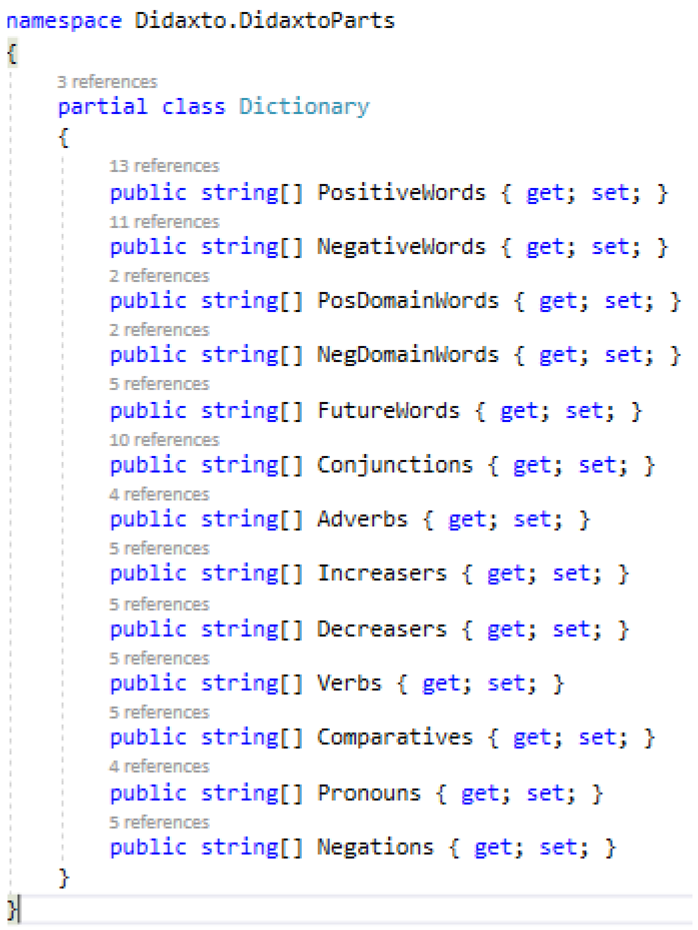
At this step the designed algorithm receives user opinions in raw form. Some form of pre-processing in order to filter-out noise was implemented. Sentence splitting is a critical step in this module (opinion delimitation) to propagate sentiment. As we said in previous chapter, we must teach our software how to read sentences. Didaxto creators, mentioned that is a critical step to split sentences from each opinion so we can make our analysis faster and easier.

3.1.3 ***Auxiliary List Preparation:***

This is a particularly crucial step. Auxiliary wordlist comprises a series of word sets like articles, verbs, comparatives, conjunctions, decreases (e.g. “less”), increasers (e.g. “more”), negations (e.g.” not”), and pronouns. These words will constitute a main feed of the algorithm. The proposed approach utilizes this seed in the constructor of all extraction patterns.

These types of words are also critical in sentiment analysis, because they can help us to figure out the real orientation of some opinion words that are included with them in the same phrase or sentence.

Here is we represent Dictionary class that includes all the wordlists that we will need to make our sentiment analysis.



```
namespace Didaxto.DidaxtoParts
{
    3 references
    partial class Dictionary
    {
        13 references
        public string[] PositiveWords { get; set; }
        11 references
        public string[] NegativeWords { get; set; }
        2 references
        public string[] PosDomainWords { get; set; }
        2 references
        public string[] NegDomainWords { get; set; }
        5 references
        public string[] FutureWords { get; set; }
        10 references
        public string[] Conjunctions { get; set; }
        4 references
        public string[] Adverbs { get; set; }
        5 references
        public string[] Increasers { get; set; }
        5 references
        public string[] Decreasers { get; set; }
        5 references
        public string[] Verbs { get; set; }
        5 references
        public string[] Comparatives { get; set; }
        4 references
        public string[] Pronouns { get; set; }
        5 references
        public string[] Negations { get; set; }
    }
}
```

3.1.4 ***Seed Import and Filtered Seed Extraction:***

The initial Seed S of the system is a set opinion words with known polarity (“bad”, “ugly”, “wonderful”, etc). The Seed is generic, meaning that it is not domain specific. The Filtered Seed S` is the set of Seed words that also appeared in the collection of opinion documents S` = S ∩ V. In other words, we filtered-out words from the Seed that do not appear in the corpus.

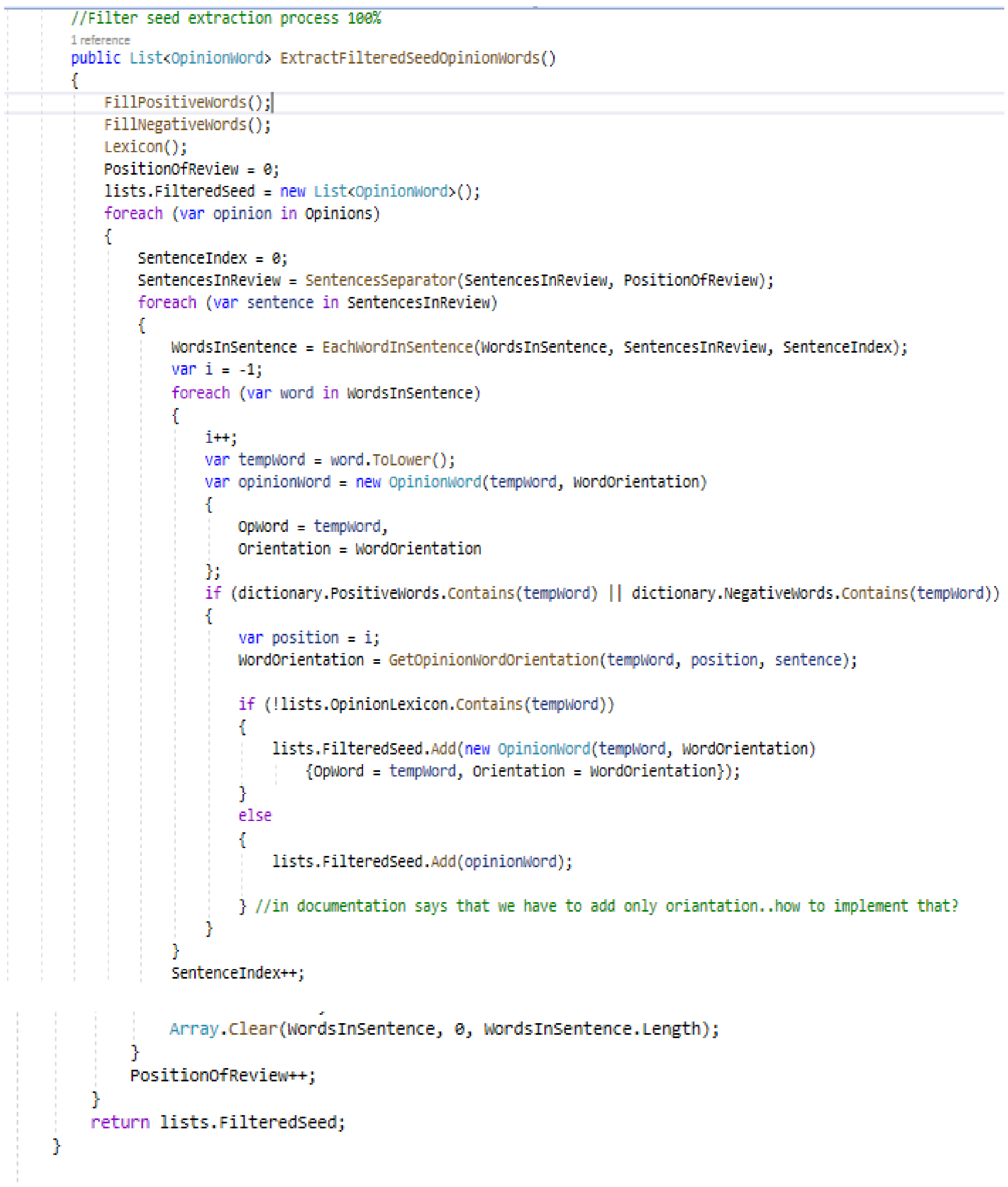
Table 2. Examples of patterns used for sentiment disambiguation

						Pol							Pol
	{fut}	{?}	{fut}	{verb}	{pos}	-1		{neg}	{verb}	{conj}	{comp}	{neg}	+1
	{neg}	{verb}	{conj}	{comp}	{pos}	-1		{neg}	{fut}	{art}	{?}	{neg}	+1
	{neg}	{pron}	{art}	{pos}	{neg}	+1		{art}	{neg}	{verb}	{decr}		+1
	{decr}	{comp}	{pos}			-1	e.g.	the	noise	has	depleted		
e.g.	little	more	useful					{neg}	{verb}	{neg}			+1
	{neg}	{pos}				-1		{pos}	{neg}				+1
	{pos}					+1		{neg}					-1

However, depending on the way the word is used, it might alter its polarity (“this phone is definitely not lightweight”). Hence, a step applied for polarity disambiguation using a set of language patterns. In table 1 some examples of polarity patterns are presented. Word abbreviations stand for {fut}: future word, {pos}: positive word, {neg}: negative word, {art}: article, {dec}: decreaser, {comp}: comparative word and {?}: any word.

The words that will be extracted at this point will be stored in Filtered Seed list.

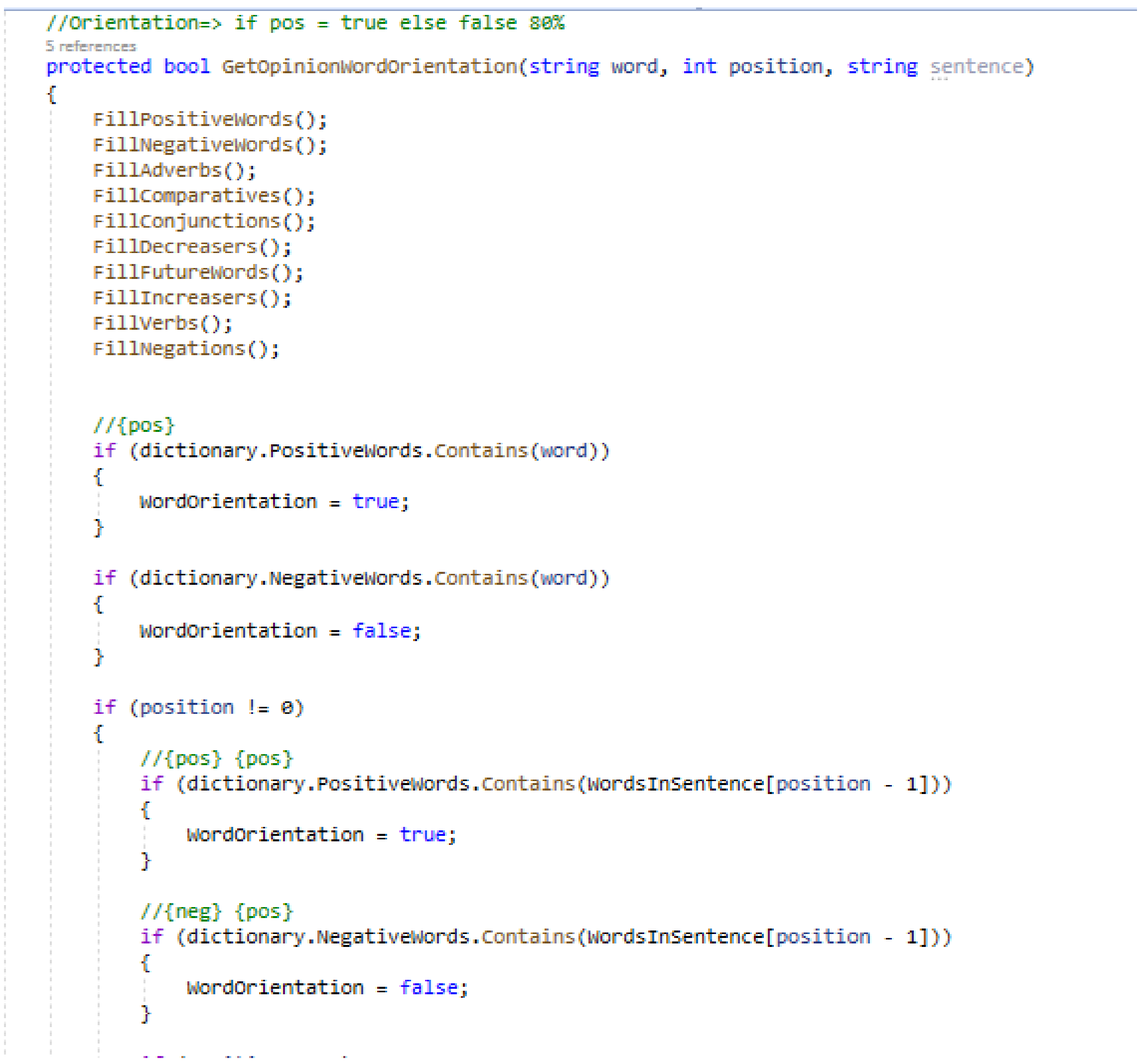
This is a function I have implemented in C# programming language, followed the logic and instructions in documentation.



```
//Filter seed extraction process 100%
1 reference
public List<OpinionWord> ExtractFilteredSeedOpinionWords()
{
    FillPositiveWords();
    FillNegativeWords();
    Lexicon();
    PositionOfReview = 0;
    lists.FilteredSeed = new List<OpinionWord>();
    foreach (var opinion in Opinions)
    {
        SentenceIndex = 0;
        SentencesInReview = SentencesSeparator(SentencesInReview, PositionOfReview);
        foreach (var sentence in SentencesInReview)
        {
            WordsInSentence = EachWordInSentence(WordsInSentence, SentencesInReview, SentenceIndex);
            var i = -1;
            foreach (var word in WordsInSentence)
            {
                i++;
                var tempWord = word.ToLower();
                var opinionWord = new OpinionWord(tempWord, WordOrientation)
                {
                    OpWord = tempWord,
                    Orientation = WordOrientation
                };
                if (dictionary.PositiveWords.Contains(tempWord) || dictionary.NegativeWords.Contains(tempWord))
                {
                    var position = i;
                    WordOrientation = GetOpinionWordOrientation(tempWord, position, sentence);

                    if (!lists.OpinionLexicon.Contains(tempWord))
                    {
                        lists.FilteredSeed.Add(new OpinionWord(tempWord, WordOrientation)
                            {OpWord = tempWord, Orientation = WordOrientation});
                    }
                    else
                    {
                        lists.FilteredSeed.Add(opinionWord);

                    } //in documentation says that we have to add only oriantation..how to implement that?
                }
            }
            SentenceIndex++;

            Array.Clear(WordsInSentence, 0, WordsInSentence.Length);
        }
        PositionOfReview++;
    }
    return lists.FilteredSeed;
}
```

Chapter 3. Didaxto

To accomplish the Filtered Seed extraction, we need to have implemented a method that returns the opinion word orientation. In documentation, refers to such a method that takes as input polarity patterns and returns Word orientation. We have implemented such method.



```
//Orientation=> if pos = true else false 80%
5 references
protected bool GetOpinionWordOrientation(string word, int position, string sentence)
{
    FillPositiveWords();
    FillNegativeWords();
    FillAdverbs();
    FillComparatives();
    FillConjunctions();
    FillDecreasers();
    FillFutureWords();
    FillIncreasers();
    FillVerbs();
    FillNegations();


    //{pos}
    if (dictionary.PositiveWords.Contains(word))
    {
        WordOrientation = true;
    }

    if (dictionary.NegativeWords.Contains(word))
    {
        WordOrientation = false;
    }

    if (position != 0)
    {
        //{pos} {pos}
        if (dictionary.PositiveWords.Contains(WordsInSentence[position - 1]))
        {
            WordOrientation = true;
        }

        //{neg} {pos}
        if (dictionary.NegativeWords.Contains(WordsInSentence[position - 1]))
        {
            WordOrientation = false;
        }
```

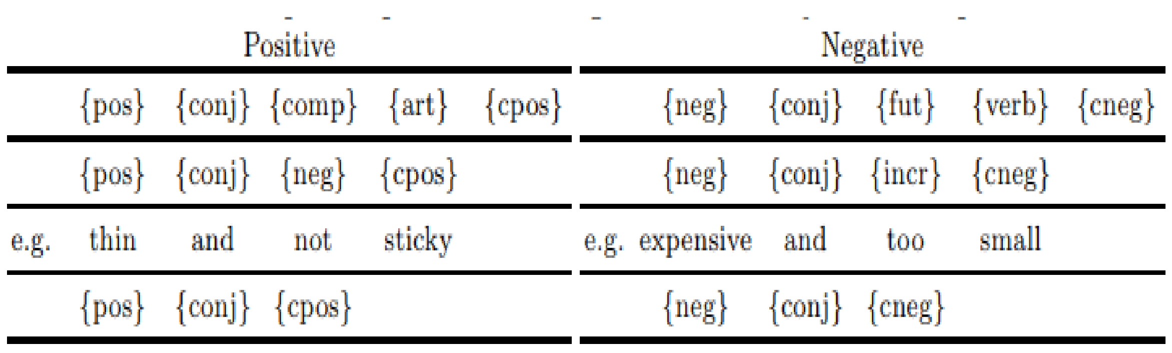
In this method we use some cases where we have to read the words that are before and after the opinion word. We try to figure out if the opinion word we find is negative, positive or neutral (Neutral orientation not implemented in this version).

In table 2 are represented some cases of words polarity.

3.1.5 ***Conjunction-Based Extraction of Opinion Words:***

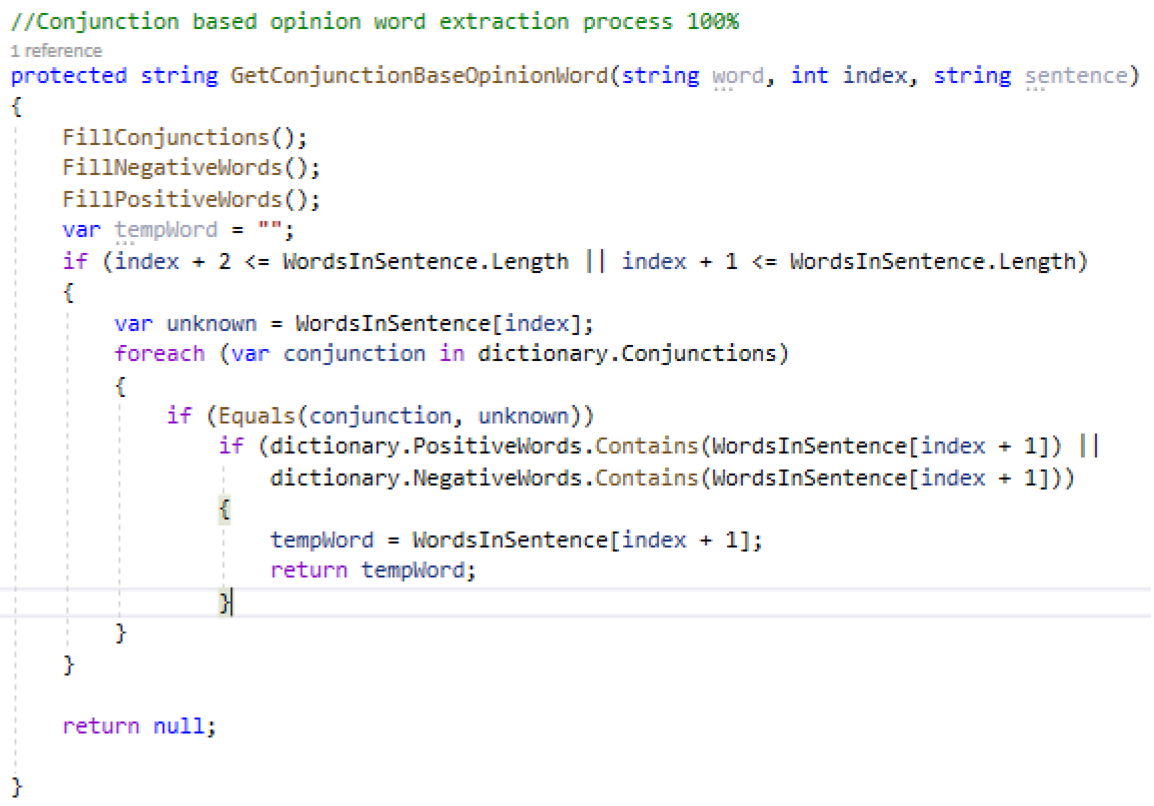
At this step we exploit the assumption of sentiment consistency that applies in conjunction words (e.g. “lightweight and well-build device”). That way “Extract conjunction words” algorithm discovers new opinion words by making use of certain conjunction patterns that have been selected to fit the sentiment consistency theories. Table 3 provide examples of such language patterns.

Table 3. Examples of positive and negative word conjunction dependencies.



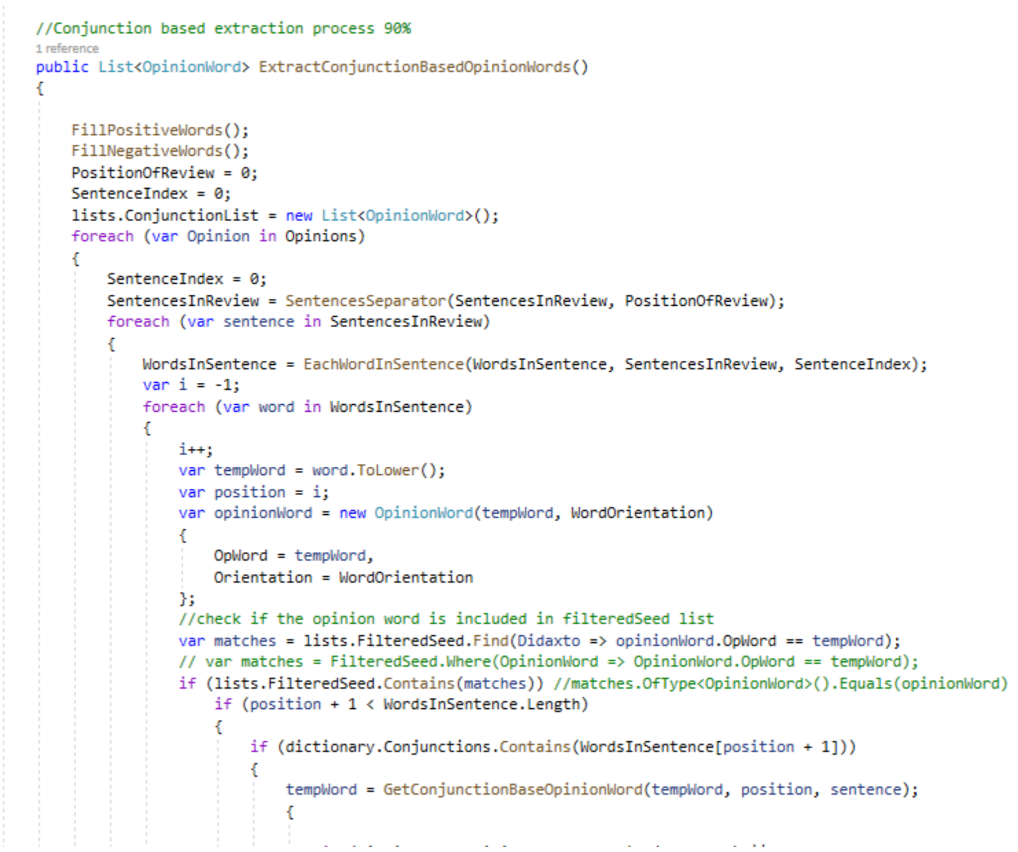
	Positive						Negative				
	{pos}	{conj}	{comp}	{art}	{cpos}		{neg}	{conj}	{fut}	{verb}	{cneg}
	{pos}	{conj}	{neg}	{cpos}			{neg}	{conj}	{incr}	{cneg}	
e.g.	thin	and	not	sticky		e.g.	expensive	and	too	small	
	{pos}	{conj}	{cpos}				{neg}	{conj}	{cneg}		

At this point we have implemented Conjunction Opinion Words Algorithm for this purpose. But first we have implemented the Conjunction based opinion word extraction process algorithm that returns (probably) an opinion word that followed by a conjunction word.

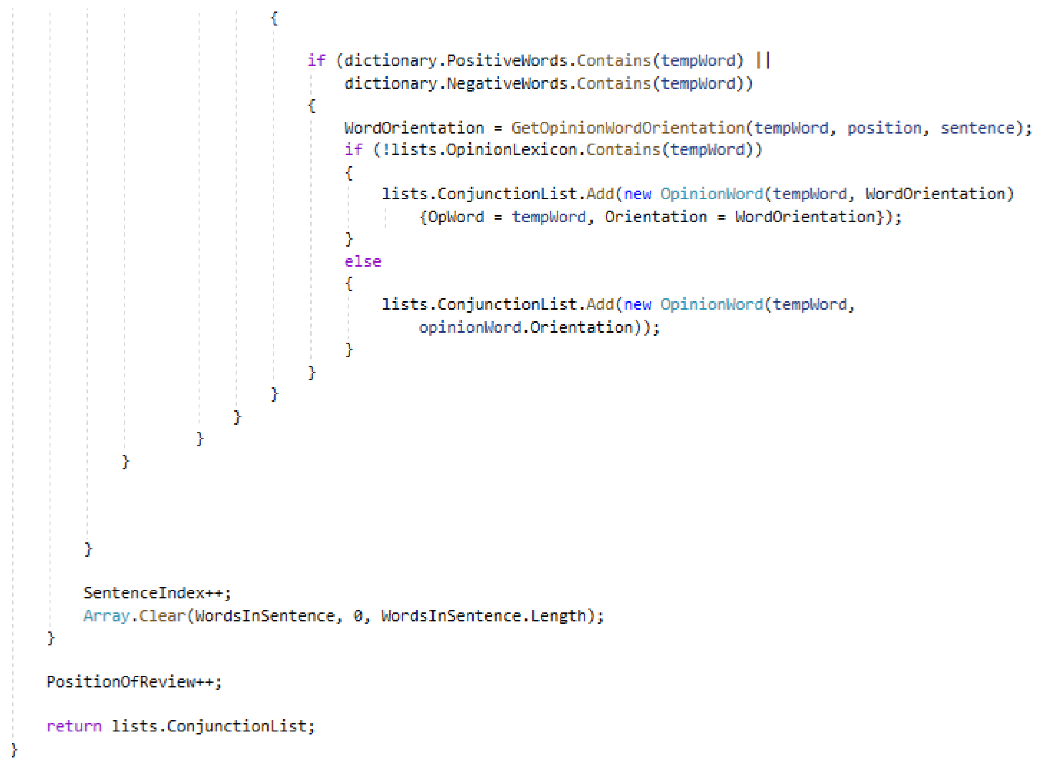


```
//Conjunction based opinion word extraction process 100%
protected string GetConjunctionBaseOpinionWord(string word, int index, string sentence)
{
    FillConjunctions();
    FillNegativeWords();
    FillPositiveWords();
    var tempWord = "";
    if (index + 2 <= WordsInSentence.Length || index + 1 <= WordsInSentence.Length)
    {
        var unknown = WordsInSentence[index];
        foreach (var conjunction in dictionary.Conjunctions)
        {
            if (Equals(conjunction, unknown))
                if (dictionary.PositiveWords.Contains(WordsInSentence[index + 1]) ||
                    dictionary.NegativeWords.Contains(WordsInSentence[index + 1]))
                {
                    tempWord = WordsInSentence[index + 1];
                    return tempWord;
                }
        }
    }

    return null;

}
```

<u>This method returns Conjunction Opinion Word list.</u>



```
//Conjunction based extraction process 90%
1 reference
public List<OpinionWord> ExtractConjunctionBasedOpinionWords()
{

    FillPositiveWords();
    FillNegativeWords();
    PositionOfReview = 0;
    SentenceIndex = 0;
    lists.ConjunctionList = new List<OpinionWord>();
    foreach (var Opinion in Opinions)
    {
        SentenceIndex = 0;
        SentencesInReview = SentencesSeparator(SentencesInReview, PositionOfReview);
        foreach (var sentence in SentencesInReview)
        {
            WordsInSentence = EachWordInSentence(WordsInSentence, SentencesInReview, SentenceIndex);
            var i = -1;
            foreach (var word in WordsInSentence)
            {
                i++;
                var tempWord = word.ToLower();
                var position = i;
                var opinionWord = new OpinionWord(tempWord, WordOrientation)
                {
                    OpWord = tempWord,
                    Orientation = WordOrientation
                };
                //check if the opinion word is included in filteredSeed list
                var matches = lists.FilteredSeed.Find(Didaxto => opinionWord.OpWord == tempWord);
                // var matches = FilteredSeed.Where(OpinionWord => OpinionWord.OpWord == tempWord);
                if (lists.FilteredSeed.Contains(matches)) //matches.OfType<OpinionWord>().Equals(opinionWord)
                    if (position + 1 < WordsInSentence.Length)
                    {
                        if (dictionary.Conjunctions.Contains(WordsInSentence[position + 1]))
                        {
                            tempWord = GetConjunctionBaseOpinionWord(tempWord, position, sentence);
                            {
```



```
                                {
                                    if (dictionary.PositiveWords.Contains(tempWord) ||
                                        dictionary.NegativeWords.Contains(tempWord))
                                    {
                                        WordOrientation = GetOpinionWordOrientation(tempWord, position, sentence);
                                        if (!lists.OpinionLexicon.Contains(tempWord))
                                        {
                                            lists.ConjunctionList.Add(new OpinionWord(tempWord, WordOrientation)
                                                {OpWord = tempWord, Orientation = WordOrientation});
                                        }
                                        else
                                        {
                                            lists.ConjunctionList.Add(new OpinionWord(tempWord,
                                                opinionWord.Orientation));
                                        }
                                    }
                                }
                            }
                        }
                    }


            }

            SentenceIndex++;
            Array.Clear(WordsInSentence, 0, WordsInSentence.Length);
        }

        PositionOfReview++;

        return lists.ConjunctionList;
    }
```

*3.1.6 **Double Propagation Extracted Opinion Words:***

This process of detecting new opinion words follows the theory of double propagation. The assumption is that each opinion word has an opinion target attached to it (e.g. “good voice”). Here, there is a direct connection (opinion word, “good”) -> (opinion target, “voice”). Based on double propagation and using the current list of opinion words, we can extract new opinion words following the same logic. So, this process is repetitive. It iterates *i* times or as long as new opinion words are discovered.

The newly discovered opinion words are going through polarity disambiguation. At this step we take advantage of intra-sentential and inter-sentential sentiment consistency. The *instar-sentential* consistency suggests that if there are other opinion words in a sentence with known orientation, then, the newly found word will get the accumulated sentiment of these words. When there are no other known opinion words in the sentence, the *inter-sentential* assumption is applied. It suggests that users tend to follow a certain opinion orientation at succeeded sentences when forming an opinion. This way if a sentence does not have an accumulated sentiment we search at nearby sentences and we assign to the new word, the largest sentiment score found at these nearby sentences. At the end of this process we have a pool of new opinion words and their orientation, the double propagation opinion words.

We employ two thresholds, the *sentiment thresholds (σ)* and the *frequency threshold (θ).* If a newly found word exceeds these two thresholds, then we consider it an opinion word. We consider opinion words only those that: 1) appear more that *θ* times along with an opinion target and 2) the sentiment polarity calculated is larger than s.

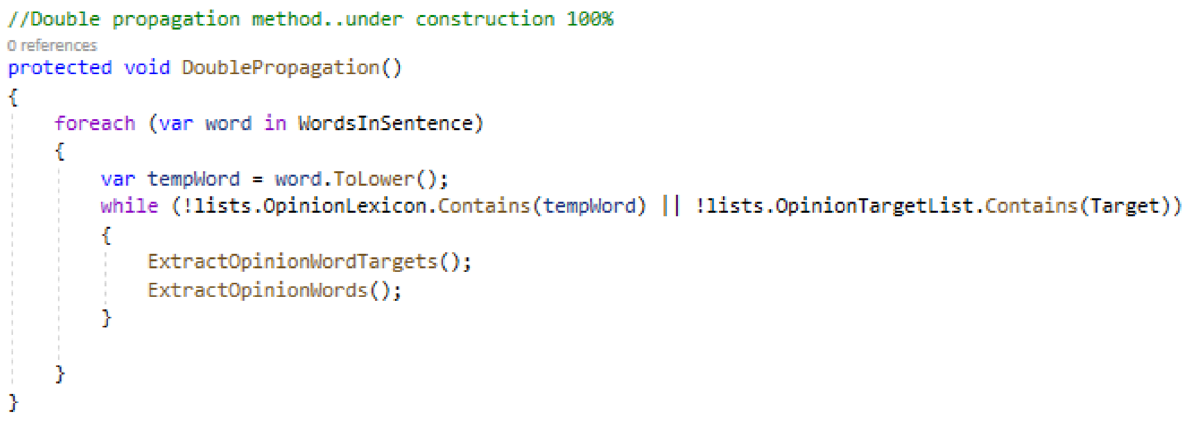
$$w_i = \begin{cases} w_i, \text{ if } \sum_{opinion=0}^{n} (w_i \wedge t_i) \geq \theta \\ \emptyset, \ \text{Otherwise} \end{cases}$$

$$[Sent]_i = \begin{cases} [Sent]_i, \text{ if Abs}([Sent]_i) \geq \sigma \\ \emptyset, \qquad \text{Otherwise} \end{cases}$$

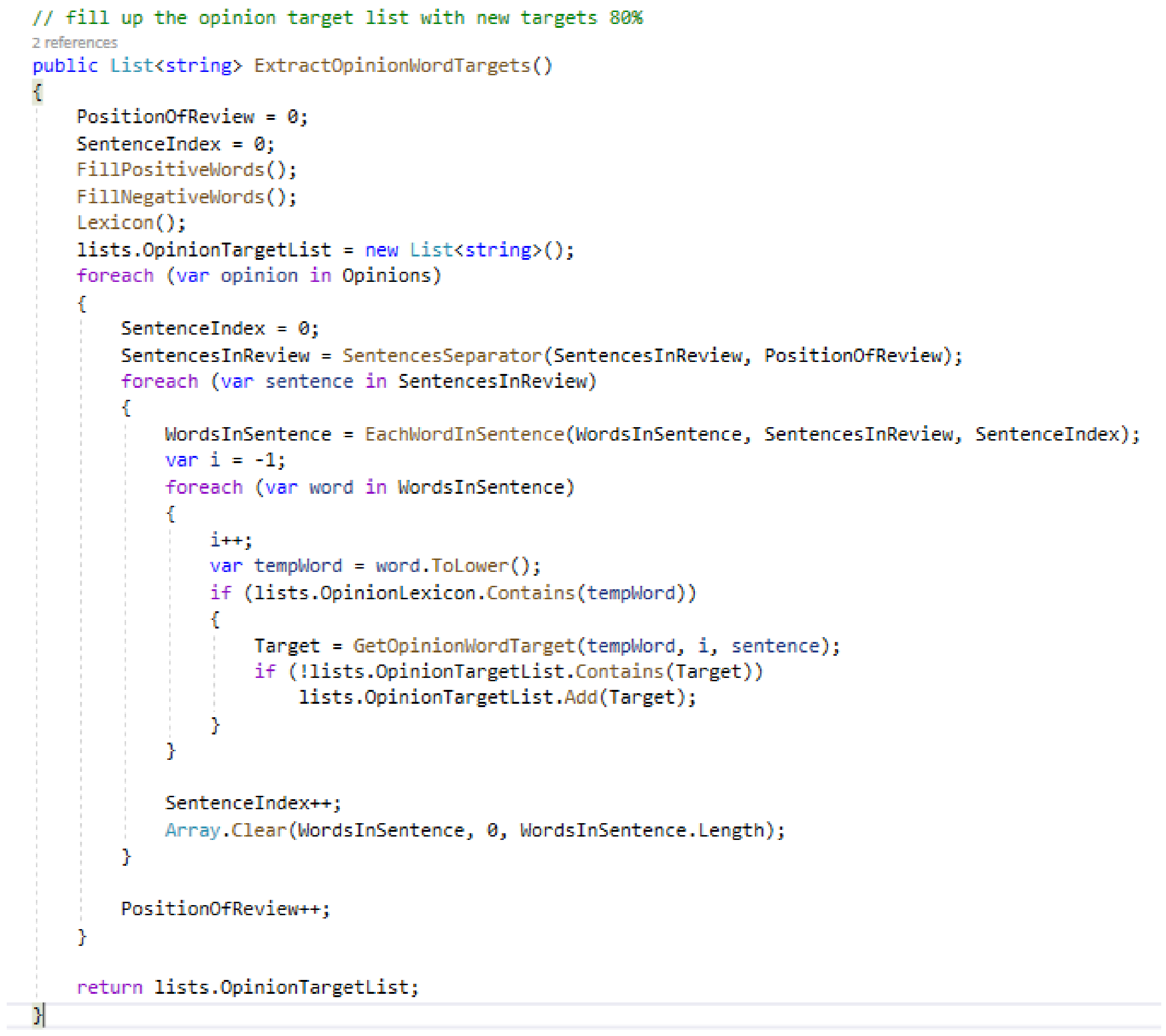
Where Wi stands for double propagation opinion word and Ti for opinion target word. Parameters θ and σ are user defined. The higher these thresholds are, we get higher precision and lower recall.

The rest algorithms are referring to double propagation approach.

Double Propagation



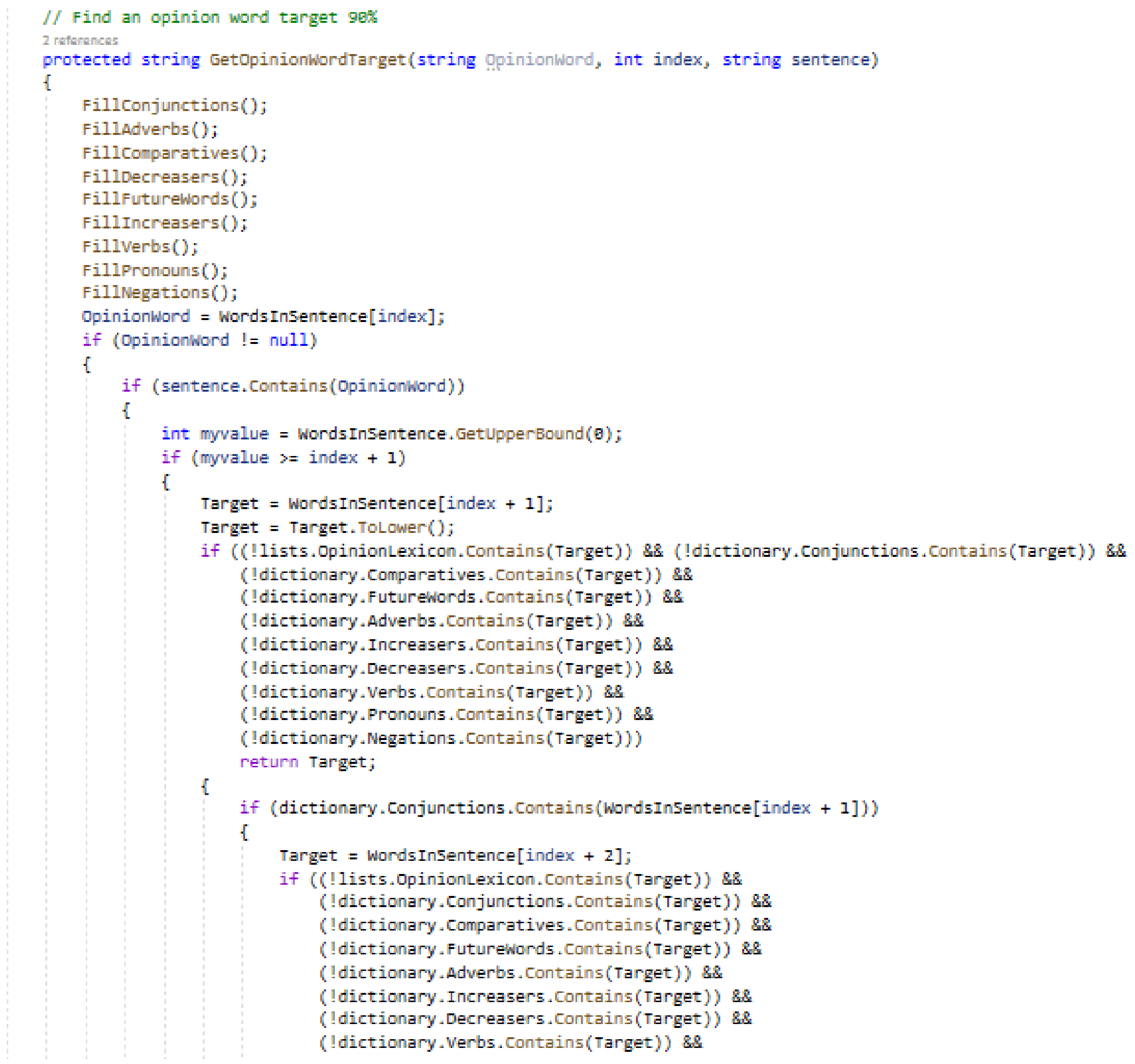
```
//Double propagation method..under construction 100%
0 references
protected void DoublePropagation()
{
    foreach (var word in WordsInSentence)
    {
        var tempWord = word.ToLower();
        while (!lists.OpinionLexicon.Contains(tempWord) || !lists.OpinionTargetList.Contains(Target))
        {
            ExtractOpinionWordTargets();
            ExtractOpinionWords();
        }

    }
}
```



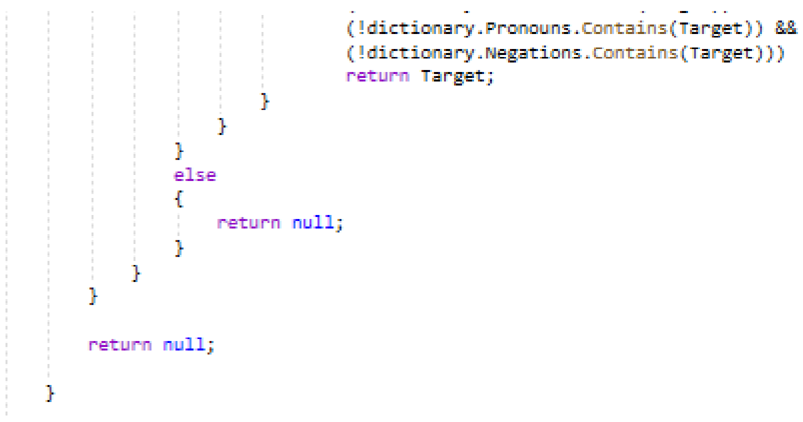
```
// fill up the opinion target list with new targets 80%
2 references
public List<string> ExtractOpinionWordTargets()
{
    PositionOfReview = 0;
    SentenceIndex = 0;
    FillPositiveWords();
    FillNegativeWords();
    Lexicon();
    lists.OpinionTargetList = new List<string>();
    foreach (var opinion in Opinions)
    {
        SentenceIndex = 0;
        SentencesInReview = SentencesSeparator(SentencesInReview, PositionOfReview);
        foreach (var sentence in SentencesInReview)
        {
            WordsInSentence = EachWordInSentence(WordsInSentence, SentencesInReview, SentenceIndex);
            var i = -1;
            foreach (var word in WordsInSentence)
            {
                i++;
                var tempWord = word.ToLower();
                if (lists.OpinionLexicon.Contains(tempWord))
                {
                    Target = GetOpinionWordTarget(tempWord, i, sentence);
                    if (!lists.OpinionTargetList.Contains(Target))
                        lists.OpinionTargetList.Add(Target);
                }
            }

            SentenceIndex++;
            Array.Clear(WordsInSentence, 0, WordsInSentence.Length);
        }

        PositionOfReview++;
    }

    return lists.OpinionTargetList;
}
```

Chapter 3. Didaxto

Opinion word to opinion target extraction process.

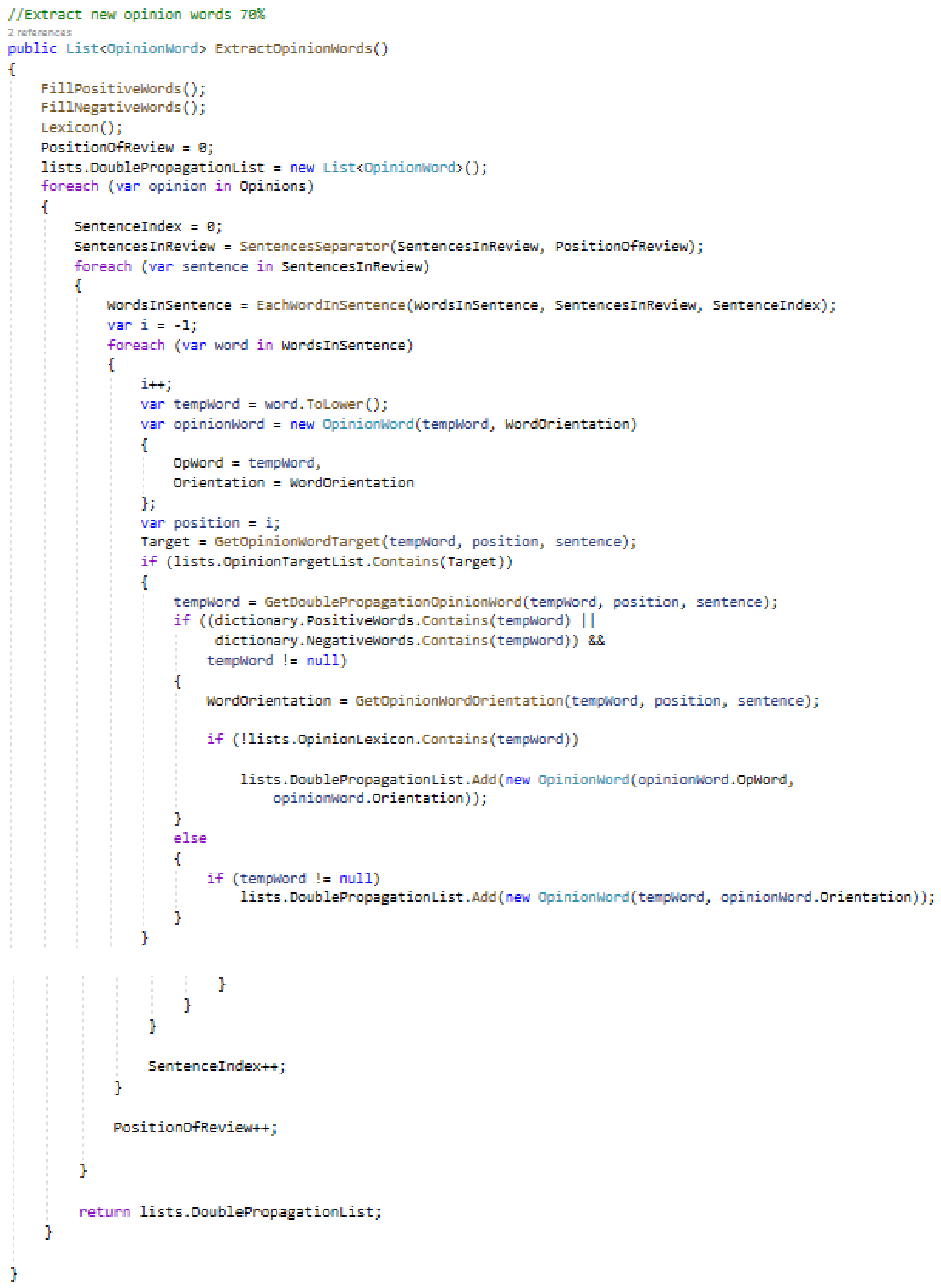


```
// Find an opinion word target 90%
2 references
protected string GetOpinionWordTarget(string OpinionWord, int index, string sentence)
{
    FillConjunctions();
    FillAdverbs();
    FillComparatives();
    FillDecreasers();
    FillFutureWords();
    FillIncreasers();
    FillVerbs();
    FillPronouns();
    FillNegations();
    OpinionWord = WordsInSentence[index];
    if (OpinionWord != null)
    {
        if (sentence.Contains(OpinionWord))
        {
            int myvalue = WordsInSentence.GetUpperBound(0);
            if (myvalue >= index + 1)
            {
                Target = WordsInSentence[index + 1];
                Target = Target.ToLower();
                if ((!lists.OpinionLexicon.Contains(Target)) && (!dictionary.Conjunctions.Contains(Target)) &&
                    (!dictionary.Comparatives.Contains(Target)) &&
                    (!dictionary.FutureWords.Contains(Target)) &&
                    (!dictionary.Adverbs.Contains(Target)) &&
                    (!dictionary.Increasers.Contains(Target)) &&
                    (!dictionary.Decreasers.Contains(Target)) &&
                    (!dictionary.Verbs.Contains(Target)) &&
                    (!dictionary.Pronouns.Contains(Target)) &&
                    (!dictionary.Negations.Contains(Target)))
                    return Target;
                {
                    if (dictionary.Conjunctions.Contains(WordsInSentence[index + 1]))
                    {
                        Target = WordsInSentence[index + 2];
                        if ((!lists.OpinionLexicon.Contains(Target)) &&
                            (!dictionary.Conjunctions.Contains(Target)) &&
                            (!dictionary.Comparatives.Contains(Target)) &&
                            (!dictionary.FutureWords.Contains(Target)) &&
                            (!dictionary.Adverbs.Contains(Target)) &&
                            (!dictionary.Increasers.Contains(Target)) &&
                            (!dictionary.Decreasers.Contains(Target)) &&
                            (!dictionary.Verbs.Contains(Target)) &&
```



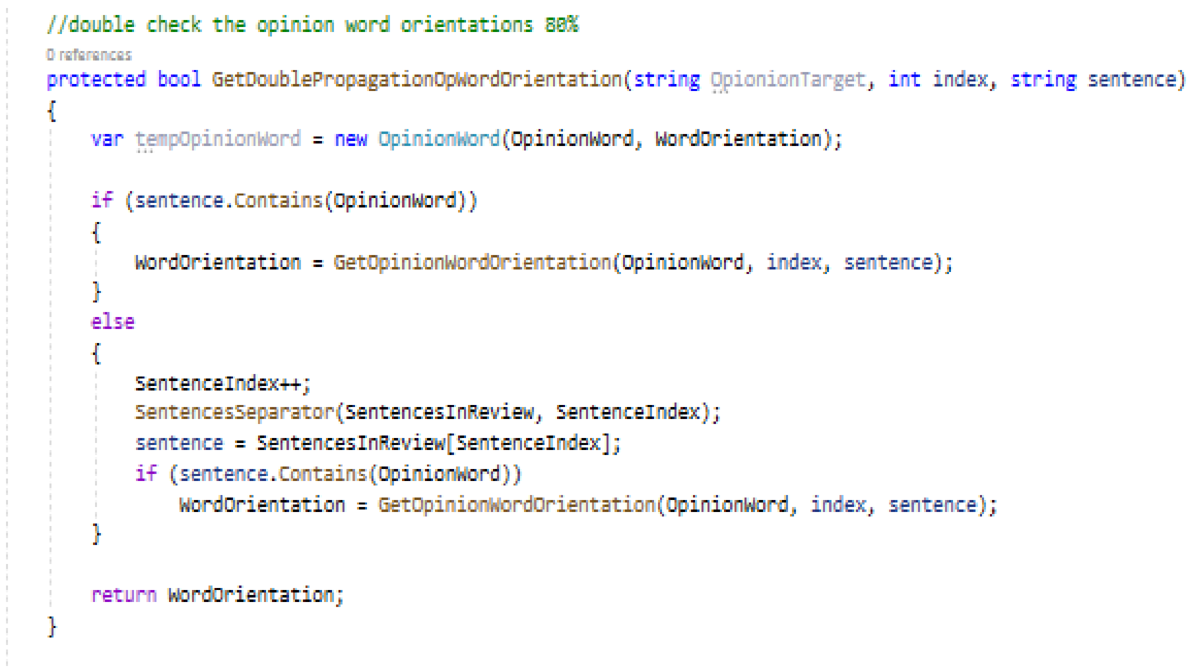
```
                            (!dictionary.Pronouns.Contains(Target)) &&
                            (!dictionary.Negations.Contains(Target)))
                            return Target;
                    }
                }
            }
            else
            {
                return null;
            }
        }
    }

    return null;

}
```

Double propagation opinion word List extraction process.



```
//Extract new opinion words 70%
2 references
public List<OpinionWord> ExtractOpinionWords()
{
    FillPositiveWords();
    FillNegativeWords();
    Lexicon();
    PositionOfReview = 0;
    lists.DoublePropagationList = new List<OpinionWord>();
    foreach (var opinion in Opinions)
    {
        SentenceIndex = 0;
        SentencesInReview = SentencesSeparator(SentencesInReview, PositionOfReview);
        foreach (var sentence in SentencesInReview)
        {
            WordsInSentence = EachWordInSentence(WordsInSentence, SentencesInReview, SentenceIndex);
            var i = -1;
            foreach (var word in WordsInSentence)
            {
                i++;
                var tempWord = word.ToLower();
                var opinionWord = new OpinionWord(tempWord, WordOrientation)
                {
                    OpWord = tempWord,
                    Orientation = WordOrientation
                };
                var position = i;
                Target = GetOpinionWordTarget(tempWord, position, sentence);
                if (lists.OpinionTargetList.Contains(Target))
                {
                    tempWord = GetDoublePropagationOpinionWord(tempWord, position, sentence);
                    if ((dictionary.PositiveWords.Contains(tempWord) ||
                          dictionary.NegativeWords.Contains(tempWord)) &&
                        tempWord != null)
                    {
                        WordOrientation = GetOpinionWordOrientation(tempWord, position, sentence);

                        if (!lists.OpinionLexicon.Contains(tempWord))

                            lists.DoublePropagationList.Add(new OpinionWord(opinionWord.OpWord,
                                opinionWord.Orientation));
                    }
                    else
                    {
                        if (tempWord != null)
                            lists.DoublePropagationList.Add(new OpinionWord(tempWord, opinionWord.Orientation));
                    }
                }


                        }
                    }
                }

                SentenceIndex++;
            }

            PositionOfReview++;

        }

        return lists.DoublePropagationList;
    }

}
```

Double propagation polarity extraction process.



```
//double check the opinion word orientations 80%
0 references
protected bool GetDoublePropagationOpWordOrientation(string OpionionTarget, int index, string sentence)
{
    var tempOpinionWord = new OpinionWord(OpinionWord, WordOrientation);

    if (sentence.Contains(OpinionWord))
    {
        WordOrientation = GetOpinionWordOrientation(OpinionWord, index, sentence);
    }
    else
    {
        SentenceIndex++;
        SentencesSeparator(SentencesInReview, SentenceIndex);
        sentence = SentencesInReview[SentenceIndex];
        if (sentence.Contains(OpinionWord))
            WordOrientation = GetOpinionWordOrientation(OpinionWord, index, sentence);
    }

    return WordOrientation;
}
```

These are the algorithms that are implemented from me, trying to reach the didaxto`s approach. My “Didaxto” application is about to get into the point of how sentiment analysis is working “behind the scenes”. The approach of the Review Explorer was to add the Didaxto source codes as “APIs” into the project and represent the results in a friendly user interface. Didaxto must do the hard job for me, find the opinion words, that will help me to connect the relevant opinions. Review explorer, in version 1.0, has the ability to represent in web application form the reviews. The future approach is to add didaxto`s functionality in the backend of the Web Application and represent the opinion connections to the user.

User Interface

Chapter_4

Abstract

User Interface of this project was the one of two most important tasks that I had to do. The first task was to create a program and add Didaxto as an API and use its functions to make the sentiment analysis. The second task was to create a nice user-friendly interface that is going to represent the output of the analysis to the user.

4.0 We built Review Explorer project in a web application form. I decided to do in this way because I thought is something more interactive and useful for the end user. This project`s goal is to provide service to the users that looking for reviews that are going to help him/her to have clear opinion about a product. Didaxto is a tool for analysts, the people who are going to extract some useful information from it and use it for their needs. So, it was built in offline mode as a desktop application. In the first version of the Review Explorer I have tried to make this program read data from a local database (Excel file) and represent the reviews in a beautiful web interface.

4.1 *Website`s pages*

1. Home page (Review Explorer)
2. Text Analysis
3. Who we are
4. Contact us

These four pages are available in the menu bar on the top of the Website. Review Explorer button on the navigation bar it redirects the user in the home page which the page you can see in the figure below.

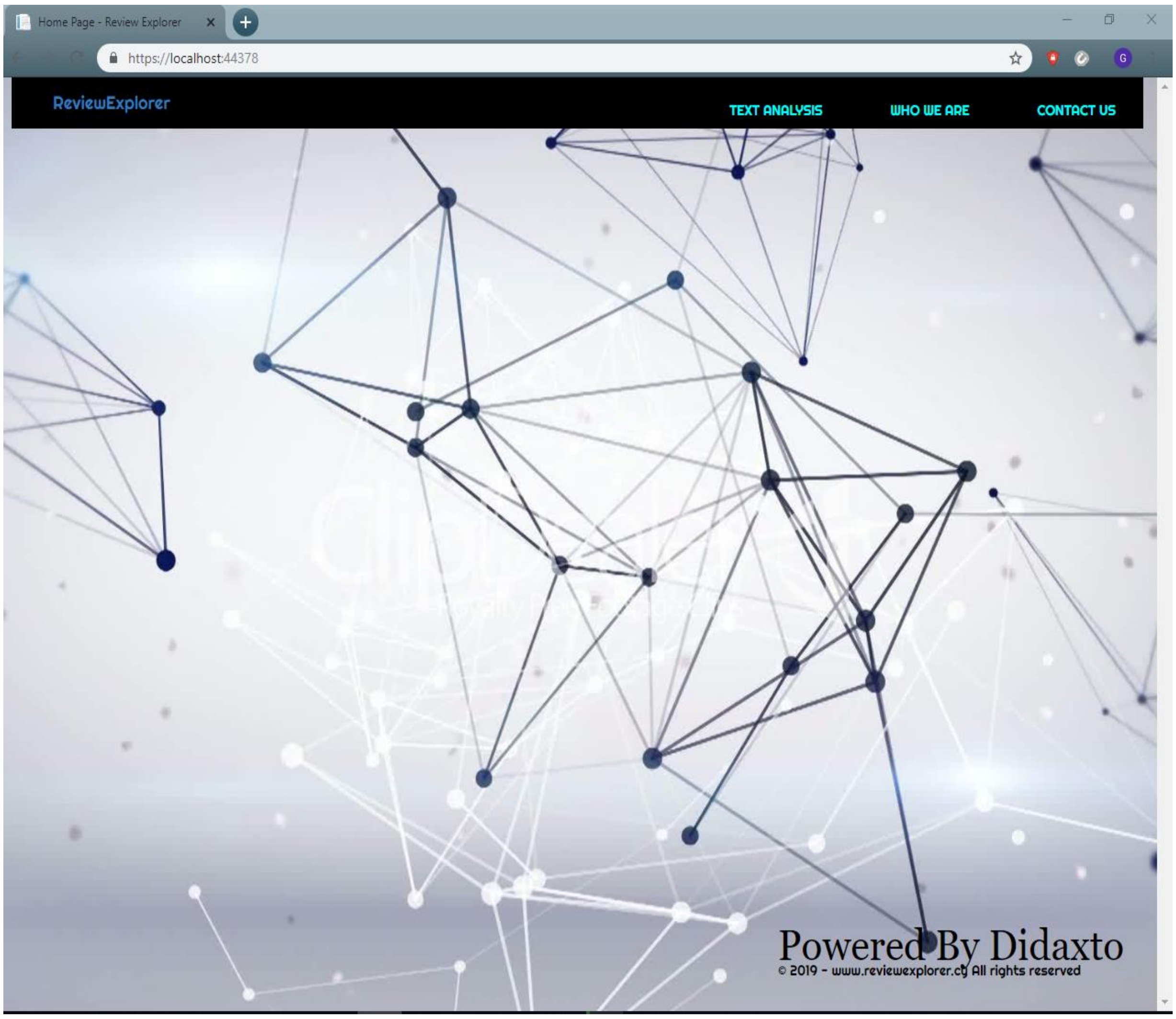


Figure 3. Home page

https://localhost:44378/home

If user clicks the “Text Analysis” tab on the navigation bar he/she will go to the Text Analysis page where we have a button (Import Review(s)). If user clicks on that button, he/she goes to a next page where two buttons are appeared.

Choose File: File explorer is open, and user should find an Excel file to upload to the page. If the file that user is selected is not an Excel file then Review Explorer is not procced and an error message is displayed.

Import: Load the selected File. User is not allowed to upload an empty file. Error message is displayed.

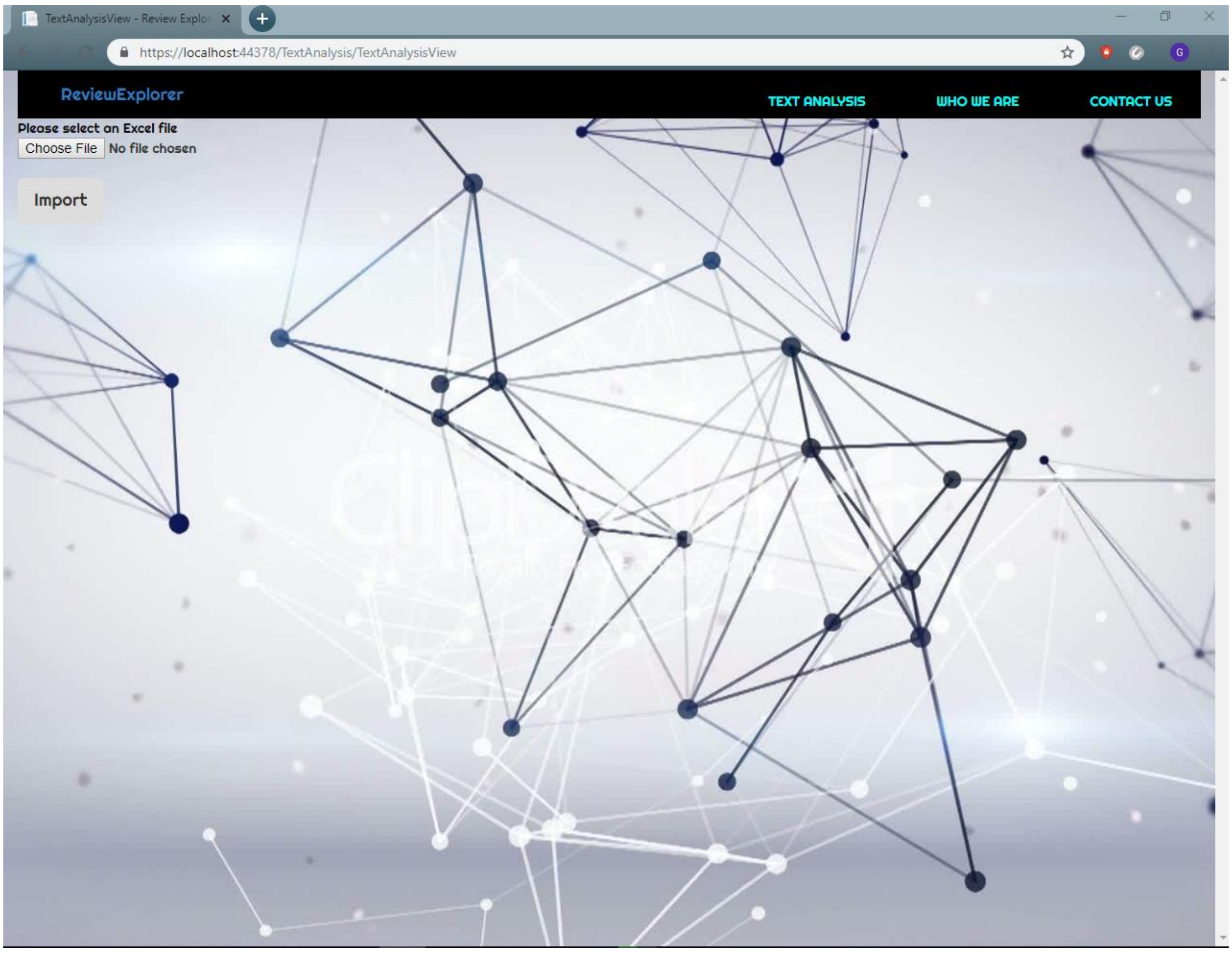

TextAnalysisView - Review Explor
https://localhost:44378/TextAnalysis/TextAnalysisView
ReviewExplorer
TEXT ANALYSIS
WHO WE ARE
CONTACT US
Please select an Excel file
Choose File
No file chosen
Import

Figure 3.1 Text Analysis Page

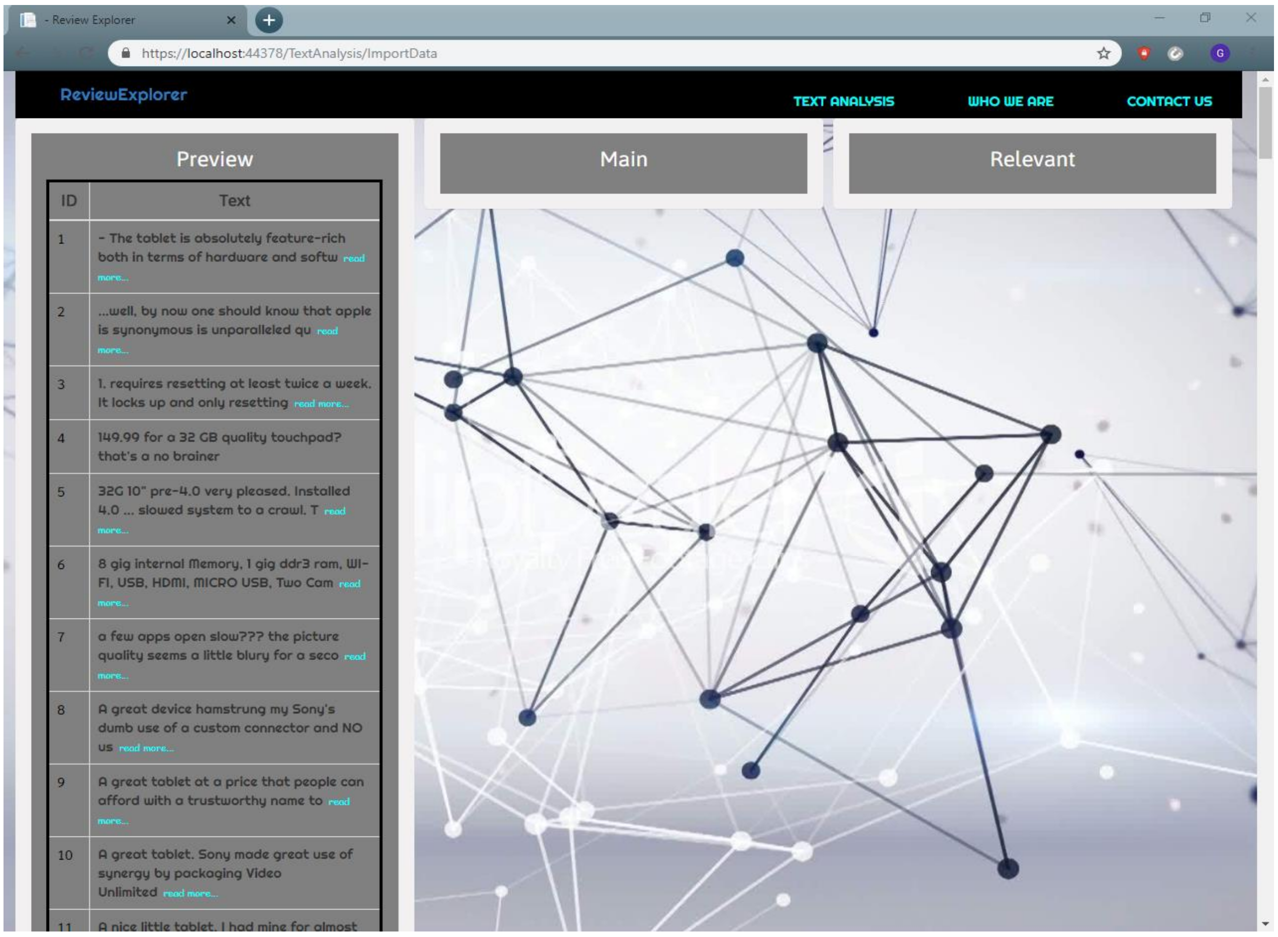
https://localhost:44378/TextAnalysis/TextAnalysisView

User successfully import an excel file with people`s opinions. In that page we have three panels, Preview, Main and Relevant.

Preview: Add each review in a table cell. Table has two columns, ID and Text where ID shows the number of each review and Text is where reviews are stored.

Main: Display the review that user has selected with key words as hyperlinks.

Relevant: Display all reviews which has the *key word that user has selected.



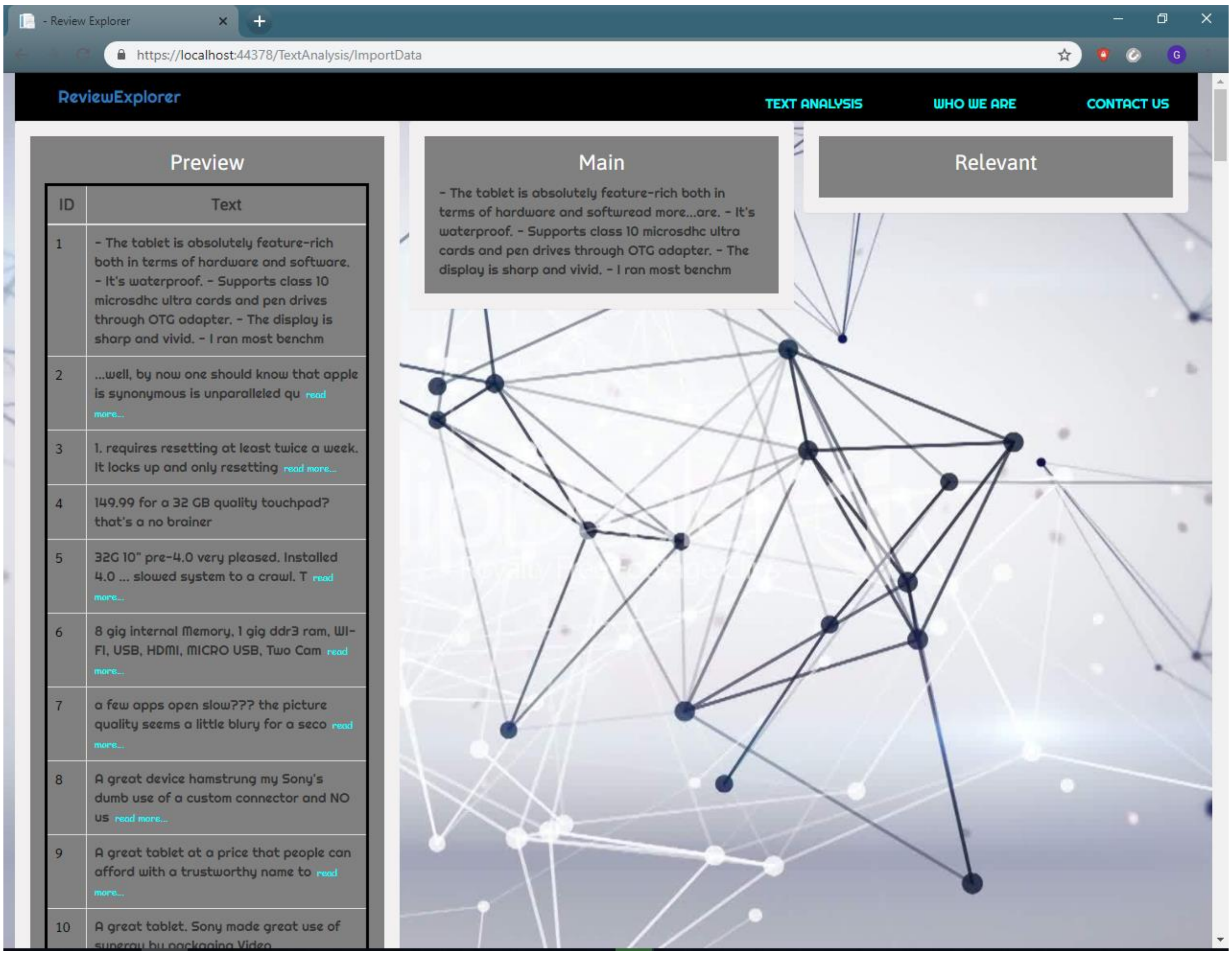
ID	Text
1	- The tablet is absolutely feature-rich both in terms of hardware and softw read more...
2	...well, by now one should know that apple is synonymous is unparalleled qu read more...
3	1. requires resetting at least twice a week. It locks up and only resetting read more...
4	149.99 for a 32 GB quality touchpad? that's a no brainer
5	32G 10" pre-4.0 very pleased. Installed 4.0 ... slowed system to a crawl. T read more...
6	8 gig internal Memory, 1 gig ddr3 ram, WI-FI, USB, HDMI, MICRO USB, Two Cam read more...
7	a few apps open slow??? the picture quality seems a little blury for a seco read more...
8	A great device hamstrung my Sony's dumb use of a custom connector and NO us read more...
9	A great tablet at a price that people can afford with a trustworthy name to read more...
10	A great tablet. Sony made great use of synergy by packaging Video Unlimited read more...
11	A nice little tablet. I had mine for almost

Figure 3.2 Reviews

https://localhost:44378/TextAnalysis/ImportData

*Key words are the opinion target words. User in an opinion is looking for the product`s attributes and the opinions about them. Attributes of a product could be words like “Screen”, “Cover”, “Size”, “Adapter”, “Touchpad” etc.

So, if user clicks on a “read more” button that review appears in the Main panel.



ID	Text
1	- The tablet is absolutely feature-rich both in terms of hardware and software. - It's waterproof. - Supports class 10 microsdhc ultra cards and pen drives through OTG adapter. - The display is sharp and vivid. - I ran most benchm
2	...well, by now one should know that apple is synonymous is unparalleled qu read more...
3	1. requires resetting at least twice a week. It locks up and only resetting read more...
4	149.99 for a 32 GB quality touchpad? that's a no brainer
5	32G 10" pre-4.0 very pleased. Installed 4.0 ... slowed system to a crawl. T read more...
6	8 gig internal Memory, 1 gig ddr3 ram, WI-FI, USB, HDMI, MICRO USB, Two Cam read more...
7	a few apps open slow??? the picture quality seems a little blury for a seco read more...
8	A great device hamstrung my Sony's dumb use of a custom connector and NO us read more...
9	A great tablet at a price that people can afford with a trustworthy name to read more...
10	A great tablet. Sony made great use of supercu by packaging Video

Main

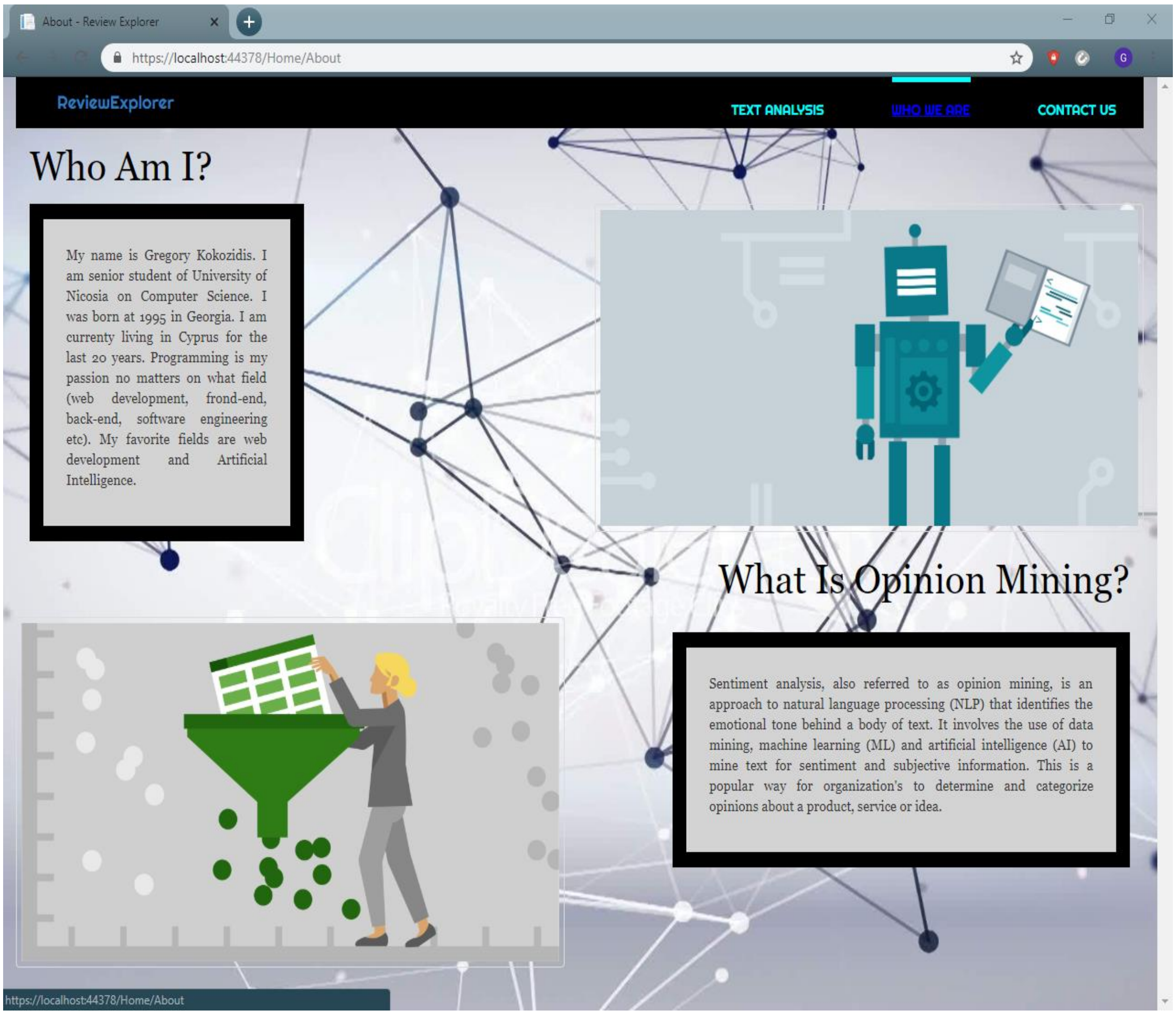
- The tablet is absolutely feature-rich both in terms of hardware and softwread more...are. - It's waterproof. - Supports class 10 microsdhc ultra cards and pen drives through OTG adapter. - The display is sharp and vivid. - I ran most benchm

Figure 3.3 Main panel

https://localhost:44378/TextAnalysis/ImportData

Chapter 4. Interface

In “Who we are” tab there are some little information about me and what is the idea behind Review Explorer.



Who Am I?

My name is Gregory Kokozidis. I am senior student of University of Nicosia on Computer Science. I was born at 1995 in Georgia. I am currenty living in Cyprus for the last 20 years. Programming is my passion no matters on what field (web development, frond-end, back-end, software engineering etc). My favorite fields are web development and Artificial Intelligence.

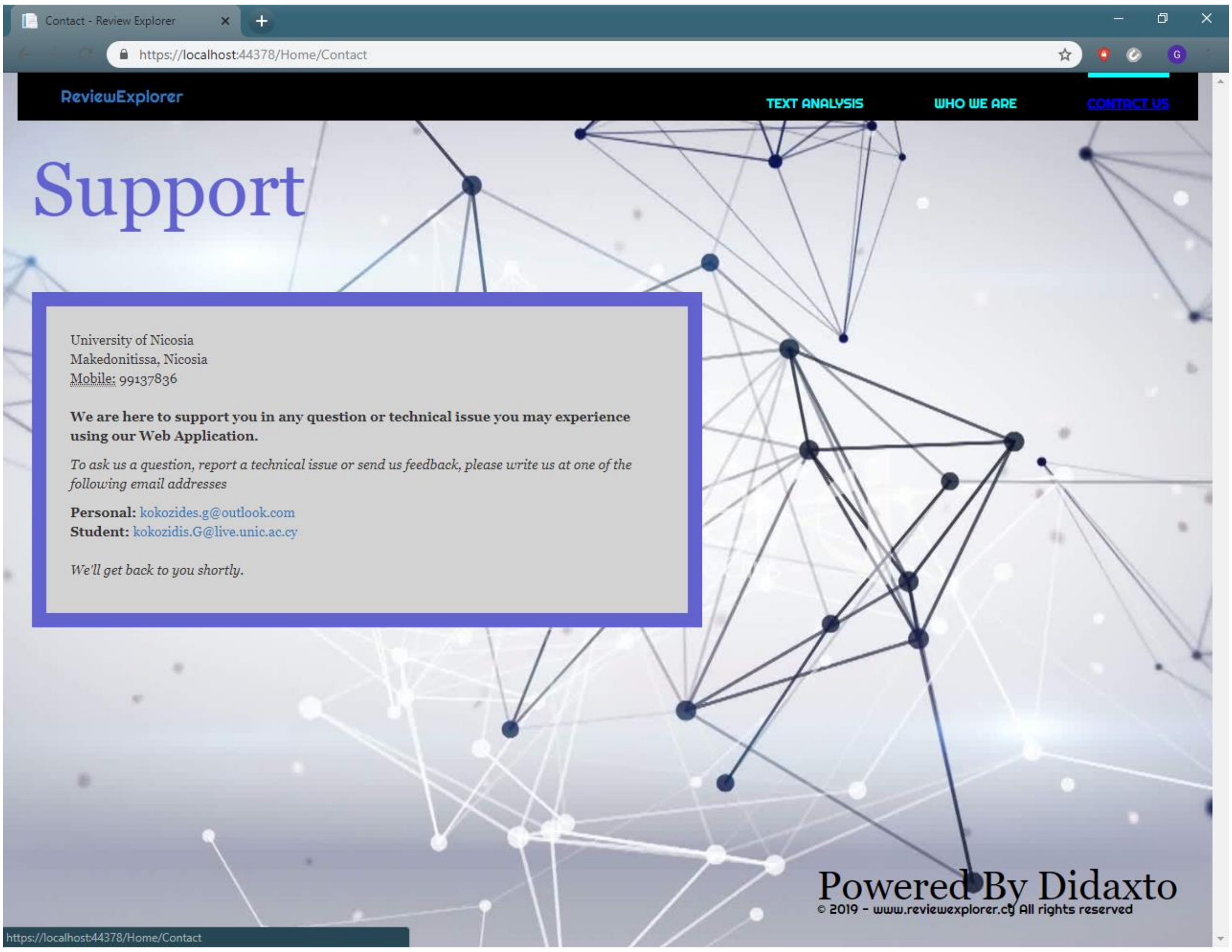
What Is Opinion Mining?

Sentiment analysis, also referred to as opinion mining, is an approach to natural language processing (NLP) that identifies the emotional tone behind a body of text. It involves the use of data mining, machine learning (ML) and artificial intelligence (AI) to mine text for sentiment and subjective information. This is a popular way for organization's to determine and categorize opinions about a product, service or idea.

Figure 3.4 Who we are page

https://localhost:44378/Home/About

In “Contact us” page is some information about where a user can leave a feedback about this Web application. This page will be useful when the Website will go live for the users (future work).


Contact - Review Explorer
https://localhost:44378/Home/Contact
ReviewExplorer
TEXT ANALYSIS
WHO WE ARE
CONTACT US

Support

University of Nicosia
Makedonitissa, Nicosia
Mobile: 99137836

We are here to support you in any question or technical issue you may experience using our Web Application.

To ask us a question, report a technical issue or send us feedback, please write us at one of the following email addresses

Personal: kokozides.g@outlook.com
Student: kokozidis.G@live.unic.ac.cy

We'll get back to you shortly.

Powered By Didaxto
© 2019 - www.reviewexplorer.cy All rights reserved

https://localhost:44378/Home/Contact

Figure 3.5 Contact us page

https://localhost:44378/Home/Contact

Conclusion and Future Work

Chapter_5

5.1 Conclusion

The aim of this thesis is to present our idea which is to connect opinions of people and make the user`s life much easier. The most critical point of this project is to apply, and user correct the tool of Didaxto. In review explorer I tried to build myself a similar software as the didaxto to get closer to the field of Artificial Intelligence and Machine Learning. Current version of this project is still pure and not so accurate on the results that it provides. This is just the first version and the begging to continue building a project that it will be inspire further NLP work.

5.1.1 Improvements

First, the Didaxto section has a lot of inaccurate outputs. That section needs a lot of work to reach the original one. And of course, the interface section must get improved to be more advanced and user friendly.

5.2 Future Work

This project to get live for the people must do some tasks perfectly. First, I will add the original didaxto`s source code in the program as APIs and add our idea to it. Didaxto implementation it will be done for personal purposes. Second, the web application will not only accept excel files as data source, but I will create a database where all dictionaries and reviews will be stored there. This database will get updated every time new data is extracted. And the last task a user will add reviews from a specific url where are reviews that he/she wants to read (amazon, ebay, online forums, facebook, twitter).

Bibliography

Chapter_6

[1] B Pang, L Lee - Foundations and Trends in Information Retrieval 2008

[2] B Lui – Synthesis lectures on human technologies, 2012

[3] A Pak, P Paroubek – LREc 2010

[4] B Liu, LZhang – Mining text data, 2012

[5] S Baccianella, A Esuli, F Sebastiani – Lrec, 2010

[6] K Ravi, V Ravi – Knowledge- Based Systems, 2015

[7] Web Information Systems Engineering- WISE 2014

[8] Pantelis Agathengelou, Ioannis Katakis Ioannis, Koutoulakis, Fotis Kokkoras, Dimitrios Gunopulos

Article Learning Patterns for discovering domain-oriented opinion words

[9] Pantelis Agathangelou Ioannis Katakis Fotios Kokkoras Konstantinos Ntonas

Mining Domain-Specific Dictionaries of Opinion Words

[10] Karin Versopoor, Kevin Bretonnel Cohen University of Colorado Denver

Natural Language Processing

[11] Diksha Khurana1, Aditya Koli1, Kiran Khatter1,2 and Sukhdev Singh1,2 1Department of Computer Science and Engineering Manav Rachna International University, Faridabad

Natural Language Processing: State of The Art, Current Trends and Challenges

[12] International Journal of Computer Applications 2012

Opinion Mining: Issues and Challenges (A survey)

6.1 Coding websites:

[1] Stack overflow

[2] Code pen

[3] Udemy

[4] Microsoft Docs

[5] Code academy

[6] GitHub

[7] W3Schools Online Web Tutorials