Bilkent University

Department of Computer Engineering

Senior Design Project

Newspector: Unreliable News Detection

Analysis Report

Ahmet Ayrancıoğlu	21601206
Deniz Dalkılıç	21601896
Kaan Gönç	21602670

Supervisor: Varol Akman

Jury Members: Halil Altay Güvenir and Özcan Öztürk

Analysis Report
November 11, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1. Introduction

People all around the world are concerned about the presence of fake news [1] for several reasons. The term "Fake News" is interpreted as "false news stories, often of a sensational nature, created to be widely shared or distributed for the purpose of generating revenue, or promoting or discrediting a public figure, political movement, company", according to the Oxford Dictionary. Spreading of fake news is a real and extremely serious threat in today's society as it can lead to violations of rights to information and expression. This in return can be used as a tool of manipulation in terms of prejudice, marginalization, excessive assimilation of ideas, acts, people, products or organizations. Furthermore, there are excessive amounts of fake news present on the internet. The main reasons for this are the ease of writing/authoring fake news; newspapers, magazines, and news agencies having their online channels, and social media being an effective medium for transmission. The internet and social media have made it very easy for anyone to publish content on a website, blog or social media profile and potentially reach large audiences. As most of the people who are reading news is getting the material from social media, many content creators use this to their advantage. These kinds of news can be encountered daily in various fields such as politics, economics, social life, security, humanity, etc. Most importantly, it is extremely hard to prevent them for 3 main reasons. Firstly, it is difficult to identify the source of the news. Secondly, it takes too much human effort and time to crosscheck and confirm their reliability. Finally, they instantly start to spread around the internet when they are published.

Even though there is a colossal amount of fake news on the internet and they may seem random and unrelated in their nature of origin, after analyzing fake news, we start to see a pattern emerge and we can roughly categorize fake news into six. These categories are clickbait, propaganda, satire, sloppy journalism, misleading headlines, and biased news.

Clickbait

Clickbaits are "material put on the Internet in order to attract attention and encourage visitors to click on a link to a particular web page". Clickbait stories use sensationalist headlines to grab attention and drive click-throughs to the publisher's website, normally at the expense of truth or accuracy [2].

Propaganda

Propagandas are ideas or statements that may be false or exaggerated and that are used in order to gain support for a political leader, party, etc. [2].

Satire

Satire is a way of criticizing a person, an idea or an institution in which you use humor to show their faults or weaknesses; a piece of writing that uses this type of criticism [2].

Sloppy Journalism

Sometimes reporters or journalists may publish a news story with unreliable information or without checking all of the facts which can in return mislead audiences [2].

Misleading Headlines

Stories that are not completely false can be distorted using misleading or sensationalist headlines. These types of news can spread quickly on social media sites where only headlines and small snippets of the full article are displayed on audience newsfeeds [2].

Biased News

Many people are drawn to news or stories that confirm their own beliefs or biases and fake news can prey on these biases. Social media news feeds tend to display news stories and articles that they think the reader will like based on gathered personal data about the reader's daily activities and searches [2].

For the final thing, in addition to the topic of fake news, when we surf on the internet we are exposed to news, articles and other content based on our own routines on the web. Such content tends to reflect our own likes, views, and beliefs and therefore isolating us from differing views and opinions present on the web. Therefore, people do not realize how beneficial it would be to access contents from different perspectives which in turn would highly increase the ability of critical thinking. Having limited access to or being unaware of the cluster of news that approaches the same topic from different perspectives is another important issue in this concept. Such a solution that defines this approach as its main methodology would provide many sources for the user to interpret and decide which source or sources are more reliable than others.[3]

Our Mission

According to media personality Hugh Linehan, “Media is no longer passively consumed – it’s created, shared, liked, commented on, attacked and defended in all sorts of different ways by hundreds of millions of people. And the algorithms used by the most powerful tech companies – Google and Facebook in particular – are brilliantly designed to personalize and tailor these services to each user’s profile.” As we particularly focus on the topic of fake news and how to increase the awareness of people against such content, we should first look at how global companies approach this problem. Leading technology companies like Google and Facebook have announced how to identify fake news with reporting and flagging tools. Furthermore, some media companies like BBC and Channel 4 are known to have their own fact-checking sites. Even though these types of movements exist against fake news, all of these processes mainly proceed manually, and therefore it takes too much time and effort to verify or deny such news. The ways for spotting fake news can be divided into different tasks:

- Checking the source of the news
- Checking your own biases
- Checking other sources for discrepancies
- Checking whether the article is a satire or not
- Reading the whole article as well as the headline
- Verifying the facts inside the article

According to our research, there is not a solid tool which successfully automates these processes of fake news detection on the market by using the linguistics analysis between multiple platforms, and we, as a team, believe that all people who encounter various news on the internet need such a tool which will interactively alert them on the unreliability factors of such news. Therefore, our mission is to design and implement such a tool and observe its impact on people’s approach to the presence of unreliable news in daily life, especially on the internet.

2. Current System

Right now we do not provide any system as we are in the research process of our project life cycle. As we have little experience in Natural Language Processing we are trying to finalize our requirements. Additionally, we are trying to gain experience in the Natural Language Processing tools and building plug-in applications using different techniques.

3. Proposed System

The main goal of *The Newspector* is to caution readers about possible unreliability and deception in the news. In order to achieve this, Newspector examines the news articles that are currently being read by the user as the user reads them and scans the other credible sources, tries to find the same or very similar claims with the original article and compares them. When it encounters any signs of a contradiction, the automated system interacts with the reader by a warning that contains the problematic claim, the reason for suspicion and the sources of the contrary claims.

The Newspector will use Natural Language Processing, Text Analysis, Sentimental Analysis algorithms to examine and make comparisons of news articles.

The platform will consist of three layers. The first one will be a web browser plugin that the user installs. The other one will be a web page that the user is directed to when unreliability is detected by the plugin. The final one is the main server that executes all the examination algorithms, analysis and comparison processes mentioned before. After the plugin extracts the article and sends it to the server, the server will make its computations and will respond back to the plugin with the possible unreliabilities. Regarding the result coming from the server, the plugin will highlight the claims that may be unreliable in the article and will show a popup that asks the user to follow a path to see the contradicted or supported articles on the Newspector web page where we will also put advertisements to support the development process of the Newspector.

3.1. Overview

The Newspector aims to help readers become more aware by gathering news articles and writings from various different sources from the internet and cross-checking them for any contradictions and reporting them to the user. To achieve this goal, the Newspector will use different approaches and tools. Our main interface with the users will be a browser plugin that will read news articles that the user is currently on and then will check the article for reliabilities using Natural Language Processing. There will also be a Newspector website where users can directly go to and browse news there and check any news they want there. Finally, there will be a Newspector backend that will handle the crosschecking and the NLP part. The requirements below will focus on these 3 parts and how the Newspector should work. However, these requirements are not finalized and can be extended further when we have more information and knowledge in the NLP field and as we continue to conduct interviews and do market research for the needs of the users.

3.2. Functional Requirements

Website

- The Newspector website should categorize news into different custom categories.

- The Newspector website should provide information about the news resources that are being used to verify different claims.

- The users should be able to search for news on desired topics.

- The users should be able to display the top charts which have the highest rated unreliable news in different categories.

- The users should be able to navigate to the source website of the news currently being examined in the Newspector website by clicking a button.

- The users should be able to suggest adding a new source website to Newspector Database.

- The users should be able to request for a specialized news comparison.

- The users should be able to give feedback about the source website.

- The users should be able to give feedback on the news given by the Newspector about the accuracy of the unreliability factor of the news.

- The Newspector Website editors should be able to manage user requests.

- The Newspector Website editors should be able to edit website information.

- The Newspector website should support advertisements, as the plug-in will not include any ads for user experience.

Plug-in

- The examination and analysis process should start working when the plug-in application is added to the web browser without needing any additional setup.

- The plug-in application should provide warnings on the user's current web page, highlighting the "unreliable" claim.

- The users should be able to see the unreliability factor of the possibly unreliable data next to the highlighted line.

- The users should be able to give feedback on the accuracy of the warning given by the Newspector.

- The users should be able to disable/enable in-text warnings through the plugin pop up window which can be accessed from the top bar.

- The users should be able to display the details for the unreliable news through the popup shown next to the highlighted line.

- The users should be able to navigate to The Newspector website through the link provided in the pop-up notification.

- The users should be able to see the old warnings through the notification icon provided on the top bar.

Server / Database

- The Newspector should be able to update its news database frequently from various sources.

- The daily activities of users should be used to improve the dataset of the Newspector.

- The news database should be backed up regularly.

- The Newspector database should store articles under related categories.

- The Newspector database should support adding articles manually.

- The Newspector database should support adding new sources to the server.

3.3. Nonfunctional Requirements

Supportability

- All classes and scripts should have a description explaining the functionality of the class and the reasoning behind its creation. Additionally, all functions should also have brief descriptions that give insight into why the developer chose to write that function.

- The project should be divided into well-defined and independent subprojects and layers in order for multiple people to work on the overall application together efficiently.

- The project should be structured in a way that would support extending the project even though some of the decisions for achieving said extendibility could be considered over-engineering by some for the initial state of the project.

Usability

- Since Newspector will be a browser plugin and will show warnings over the news, the warning should be seen with ease but should not disturb the user from reading the news article.

- The warning popup should be able to clearly show the user where the unreliability resides and should provide a button for the user to go the Newspector website for further analysis of the article.

- The Newspector should be available to download in the Chrome Web Store for ease of access.

- The plugins should have an option to show users a page with instructions and a rough explanation of how the Newspector works behind the scenes.

Reliability

- Gathering the news pieces from selected sources is crucial for the Newspector to function. So, Newspector should be able to gather news from at least 90% of the selected sources at a given time.

- To increase the trust of users towards the Newspector, false reporting of solid well supported information that can be validated by other sources should be kept at a minimum of 10%.

- The sources should be selected carefully by a knowledgeable person in order to reduce bias.

- Newspector should update its sources so that no deadlinks will ever be presented to a user such as removed news articles, etc.

- The data storage solution for the project should be selected in a way that will allow the Newspector to be functional when a crash happens on one of the servers that store the news data.

- The data storage solution for the project should support scalability and should not require the reconstruction of the database when the stored data gets too big.

Performance

- The gathering of the news from selected sources should be fast and effective to include the most up to date news for cross-validation.

- Examination of the news article being read by the user should be fast and be done in real-time as the user reads.

- The Newspector plugin should not cause any slowdown on the website the user is on.

- The servers running the cross-validation algorithms should be fast and can be auto-scaled as the population of the userbase grows to provide the same performance for every user.

Extendibility

- The addition of further news sources should be easy. Other parts of the application should not be dependent on where the news is coming from.

- The addition of further categories should not affect the rest of the cross-validation algorithms.

Portability

- The process of porting the Newspector to mobile platforms should be easy and should be connected to the same backend server.

- The user experience of the Newspector should be the same on various platforms.

Privacy

- The use of cookies and other means of gathering information should be done with the permission of the user.

- Data gathered by users should not be associated with users directly unless they give their consent.

- Data gathered should be securely stored in our databases as it can easily be used to target news or advertisement by other platforms.

3.4. Pseudo Requirements

Implementation Constraint

For the development process of the NLP and Machine Learning algorithms of the Newspector, we will use the Python programming language. The frontend application will be implemented using HTML and CSS. For data management, our plan is to use Google's Firebase platform. The development process will be controlled and managed by the git version control system.

Economical Constraint

The Newspector is a free-to-use platform and does not have a sign-up system. So the implementation process should assess the minimum amount of expense for technical tools, libraries, APIs, data samples, etc. On the other side, Newspector should try to maximize the amount of income from advertisements without losing user engagement. Most importantly, as our main goal is to provide a tool to raise awareness on the presence of fake news for each person without regarding their economic status, our plan is not to implement a premium membership system but rather use advertisements to make a profit.

Political Constraint

Since the Newspector will label some news articles as "possibly unreliable" if necessary, the user should not feel that these warnings are biased in a political manner such as supporting a specific political party or view. To achieve this feeling of objectiveness, sources of articles that will be examined to cross-check "possible unreliable" articles should be selected carefully and with no bias. For example, if two sources would be selected from one political view, two more sources should be selected from the opposing view.

Social Constraint

The Newspector should not make the user feel like the warnings raised by the platform are biased in some social manner. For instance, the users should not feel like articles marked as "possibly unreliable" are biased towards a specific ethnicity, religion or social class.

Privacy Constraint

Since personal data protection is an important issue and there are regulations about it such as General Data Protection Regulation (GDPR) by European Union (EU), the Newspector won't store or use any data related to a user. Some data can be gathered for the sake of enhancing the current system. For instance, to broaden the datasets of the machine learning algorithms used. However, none of the gathered data should be associated with a user.

3.5. System Models

3.5.1. Scenarios

Textual Use Case Scenario Descriptions for Newspector Website

In this section, only the main possible scenarios for the users are described both for The Newspector Website and The Newspector Plugin.

Description of Search News Use Case

Name: Search News

Participating actor: Newspector User

Entry condition:

- The user is currently displaying the main page of the Newspector Website.

Exit condition:

- The user successfully finds specific news headlines according to the searched keywords.

Successful Scenario Event Flow:

- The user clicks the search bar given on top of the main page of the website.
- The search bar asks for input of keywords from the user.
- The user types the desired keywords and presses the search icon.
- The website provides the news found on the database related to the given keywords to the user.

Special Requirements

- The user can use filters in their searches.

Description of Display Top Charts Use Case

Name: Display Top Charts

Participating actor: Newspector User

Entry condition:

- The user is currently displaying the main page of the Newspector Website.

Exit condition:

- The user successfully displays news headlines as categorized into different topics.

Successful Scenario Event Flow:

- The user clicks the "Top Charts" button on top of the main page.
- The user is directed to a new page.
- The website displays news headlines of news that have the highest unreliability factors in different categories.

Special Requirements

- The user can use filters while displaying top charts.

Special Requirements

- none

Description of Read News Use Case

Name: Read News

Participating actor: Newspector User

Entry condition:

- The user is currently displaying the main page of the Newspector Website.

Exit condition:

- The user successfully reads specific news.

Successful Scenario Event Flow:

- User searches for specific news headlines using keywords as shown in the "Search News" use case.
  or
- The user displays news headlines through the top charts as shown in the "Display Top Charts" use case.
- The user selects a news headline.
- The user is directed to the page where the news content is provided.

Exceptional Scenario Event Flow:

- User searches for specific news headlines using keywords as shown in the "Search News" use case.
  or
- The user displays news headlines through the top charts as shown in the "Display Top Charts" use case.
- The user selects a news headline.
- The user is directed to the page where the news content is provided.
- The user decides to give feedback on the news in terms of the unreliability factor.
- The user clicks the "Give Feedback" button.
  or
- The user decides to give feedback on the source website of the news in terms of the unreliability factor.
- The user clicks the "Give Feedback on Source Website" button.

Special Requirements

- none

Description of Navigate to News Source Use Case

Name: Navigate to News Source

Participating actor: Newspector User

Entry condition:

- The user is currently reading news through the process shown in the “Read News” use case.

Exit condition:

- The user successfully navigates to the source website of the news.

Successful Scenario Event Flow:

- The user scrolls to the end of the news to see news details.
- The user clicks the news source icon or heading.
- The user is navigated to the source website of the news currently being read.

Special Requirements

- none

Description of Suggest News Source Use Case

Name: Suggest News Source

Participating actor: Newspector User

Entry condition:

- The user is currently displaying the main page of the Newspector Website.

Exit condition:

- The user successfully makes a request for adding a new source to the Newspector Database.

Successful Scenario Event Flow:

- The user scrolls to the end of the main page where detailed info about the Newspector can be found.
- The user clicks the "Request for a New Source" label.
- A pop-up screen is displayed with a text field asking for the news source website.
- The user types the URL of the new source website and clicks "Done".
- A request for the addition of a new source website is successfully made.

Special Requirements

- none

Description of Request News Comparison Use Case

Name: Request News Comparison

Participating actor: Newspector User

Entry condition:

- The user is currently displaying the main page of the Newspector Website.

Exit condition:

- The user successfully makes a request for making a news comparison to Newspector Server.

Successful Scenario Event Flow:

- The user clicks the "Request News Comparison" button.
- The website asks for a specific input of news.
- The user attaches the document or pastes the article to the provided field.
- The user selects "Done".
- The given article is successfully sent to Newspector Server for unreliability comparison.

Special Requirements

- none

Textual Use Case Scenario Descriptions for Newspector Plug-in

Description of Read In-text Warnings Use Case

Name: Read In-text Warnings

Participating actor: Newspector User

Entry condition:

- The user is currently on a news website displaying an article.
- Newspector plug-in is currently active.

Exit condition:

- The user successfully sees the warning for the possibly unreliable news being read.

Successful Scenario Event Flow:

- Plug-in successfully highlights the possibly unreliable data in the article.
- The user finds the highlighted line in the article.
- The user reads the highlighted line containing the possibly unreliable data.

Exceptional Scenario Event Flow:

- Plug-in successfully highlights the possibly unreliable data in the article.
- The user brings the cursor on top of the underlined sentence.
- The user disables the in-text warnings using the pop-up window (or through the extension on the top bar).
  or
- The user reports the warning thinking that the unreliability factor of the article is not accurate.

Special Requirements

- none

Description of Navigate to Source Website Use Case

Name: Navigate to Source Website

Participating actor: Newspector User

Entry condition:

- The user is currently on a news website displaying an article.
- Newspector plug-in is currently active.

Exit condition:

- The user successfully navigates to the source website of the news.

Successful Scenario Event Flow:

- Plug-in successfully highlights the possibly unreliable data in the article.
- The user brings the cursor on top of the underlined sentence.
- A pop-up window is opened.
- The user clicks the website icon on the pop-up menu...
- The user is successfully directed to the source website.

Special Requirements

- none

Description of Display Unreliable News Details Use Case

Name: Display Unreliable News Details

Participating actor: Newspector User

Entry condition:

- The user is currently on a news website displaying an article.
- Newspector plug-in is currently active.

Exit condition:

- The user successfully displays the details for the possibly unreliable news being read.

Successful Scenario Event Flow:

- Plug-in successfully highlights the possibly unreliable data in the article.
- The user brings the cursor on top of the underlined sentence.
- A pop-up window is opened.
- The user clicks the details icon on the pop-up menu.
- Detailed information about the news is shown to the user.

Special Requirements

- none

Description of Display Unreliability Factor Use Case

Name: Display Unreliability Factor

Participating actor: Newspector User

Entry condition:

- The user is currently on a news website displaying an article.
- Newspector plug-in is currently active.

Exit condition:

- The user successfully displays the unreliability factor of the possibly unreliable news being read.

Successful Scenario Event Flow:

- Plug-in successfully highlights the possibly unreliable data in the article.
- The user brings the cursor on top of the underlined sentence.
- A pop-up window displaying the unreliability factor of the news is opened.

Special Requirements

- none

Description of Display Old Warnings Use Case

Name: Display Old Warnings

Participating actor: Newspector User

Entry condition:

- The user is currently on a news website displaying an article.
- Newspector plug-in is currently active.

Exit condition:

- The user successfully displays the old warnings on possibly unreliable news.

Successful Scenario Event Flow:

- The user clicks the extension icon located on the top bar of the browser.
- A pop-up window is opened.
- The user clicks the notification icon.
- Old warnings on possible unreliable news are shown to the user.

Special Requirements

- none

3.5.2. Use Case Model

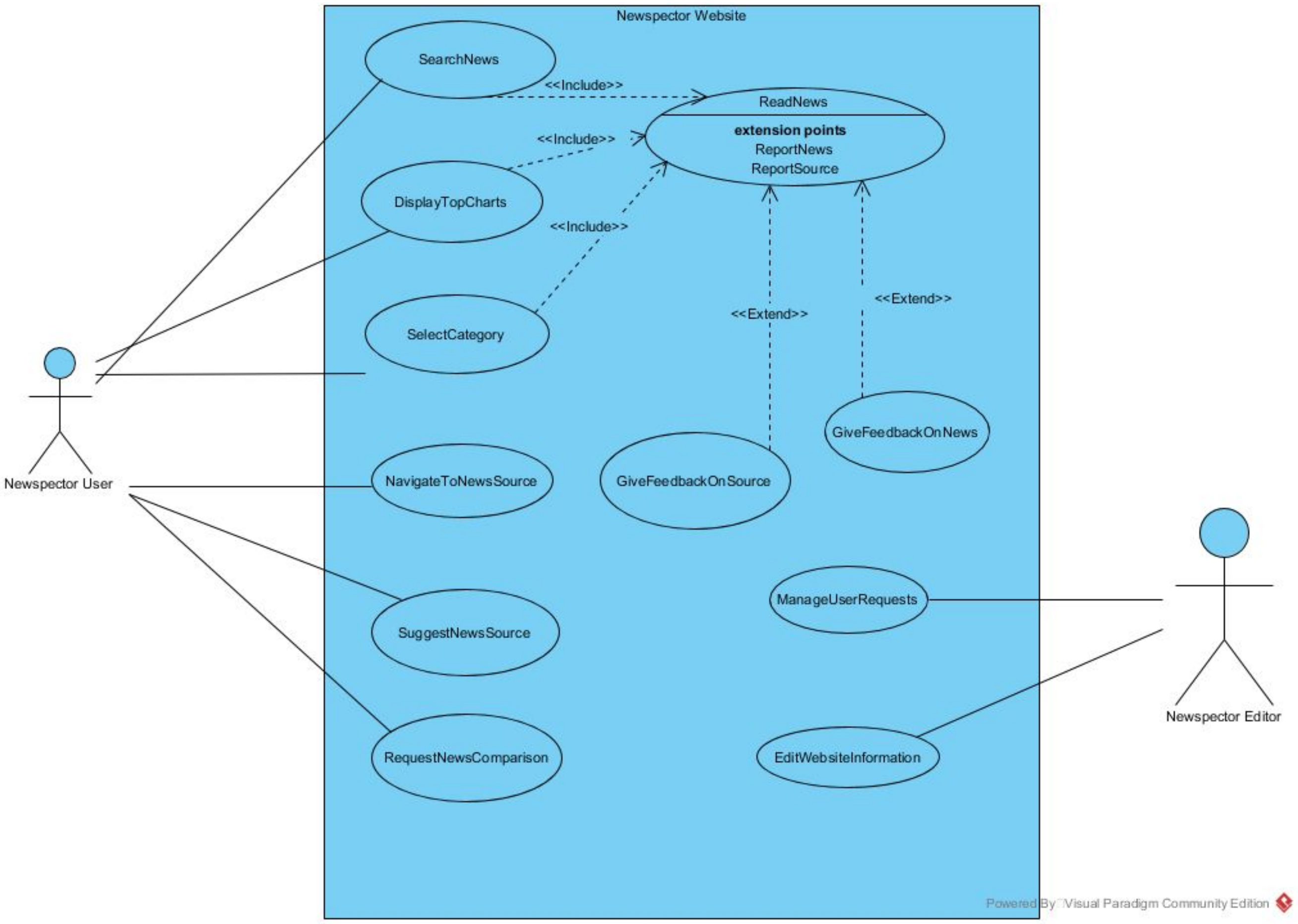


Fig. 2

There are two main actors who operate on the system of the Newspector Website, a user, who interacts with the provided news and an editor, who manages the requests of the users and maintenance of the website. The main use cases for these two actors are given in the use case diagram above which are consistent with the functional requirements of the website given in part 3.2.

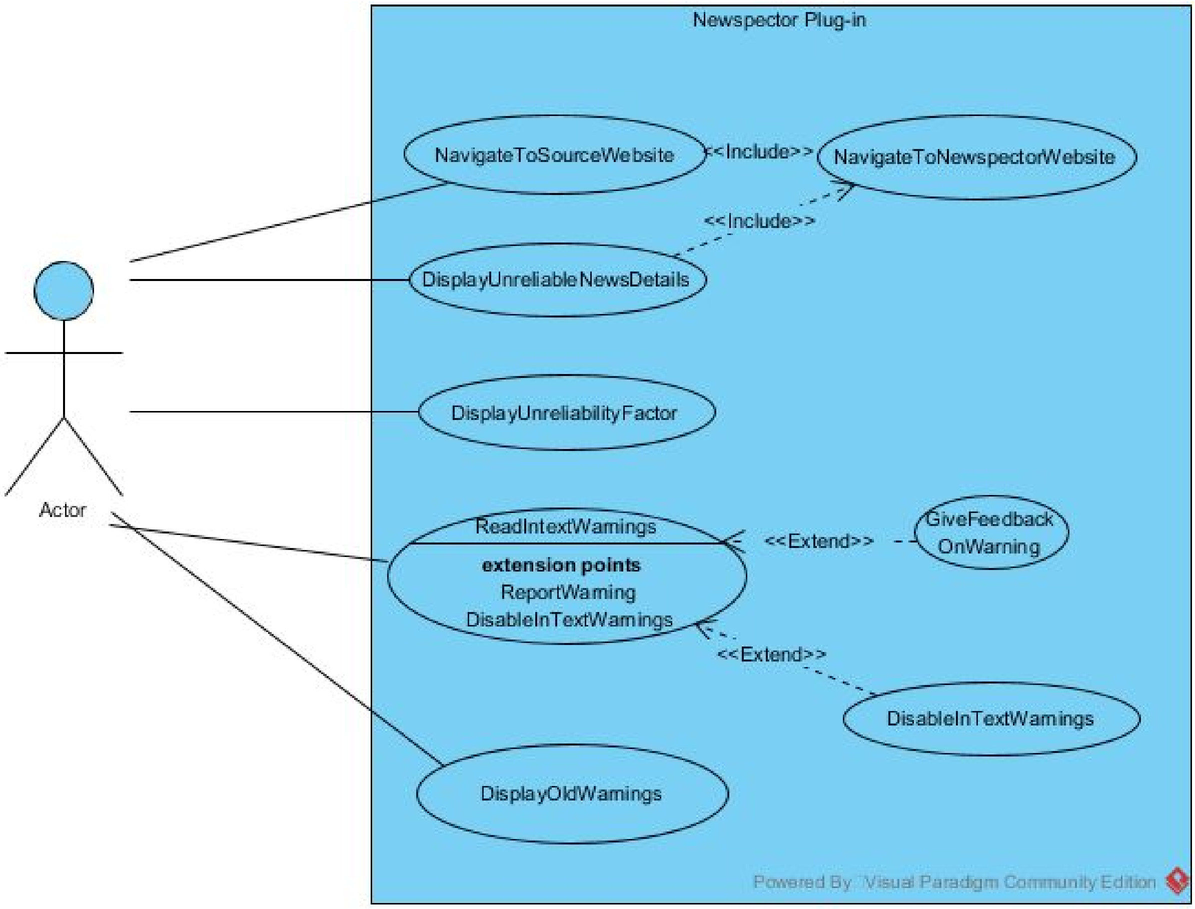


Fig. 3

There is only one actor who operates on the system of The Newspector Plugin, a user, who interacts with the provided news and the related warnings displayed by the plugin. The main use cases for the user are given in the use case diagram above which are consistent with the functional requirements of the plugin application given in part 3.2.

3.5.3. Object and Class Model

The total object diagram can be seen in Figure X. In this section, this diagram will be divided into sub-diagrams and the roles of the objects and the relationships between them will be explained clearly.

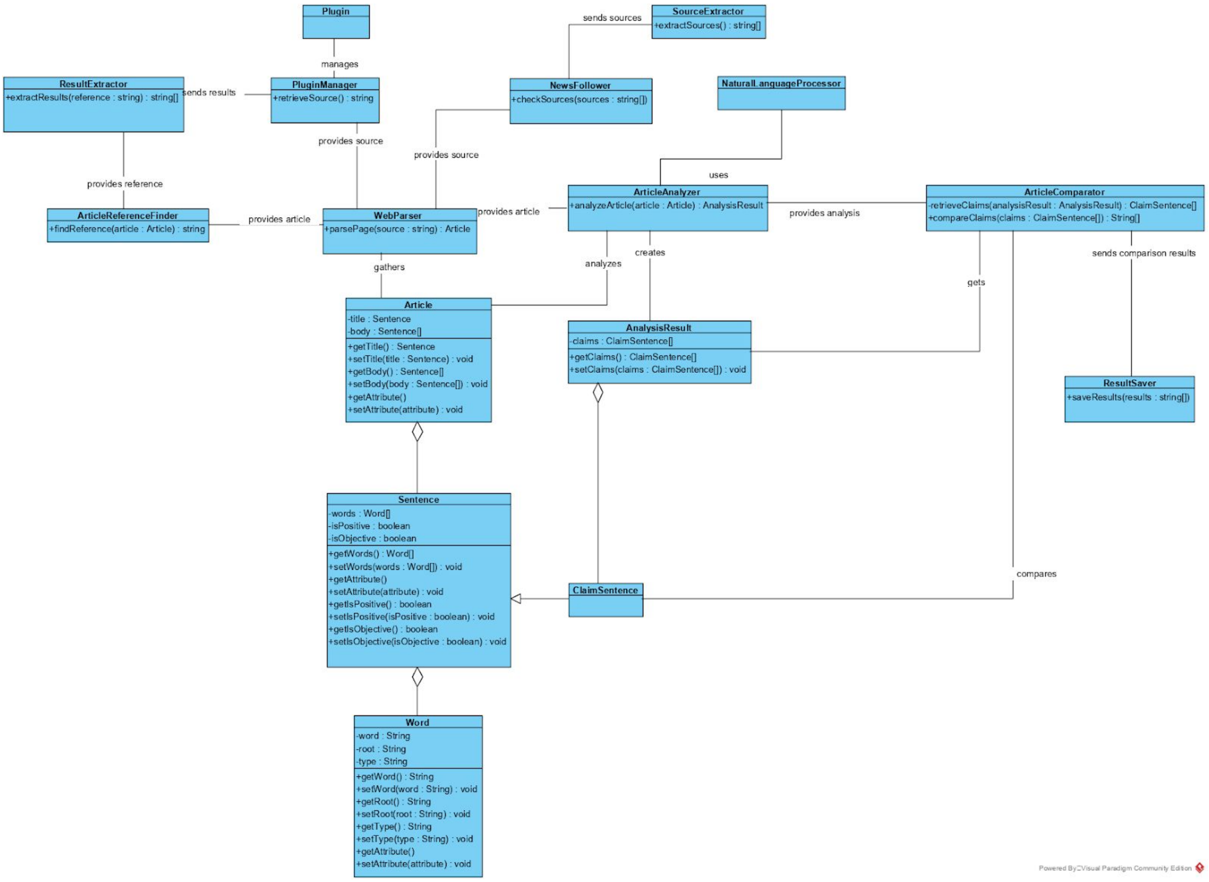

Plugin
manages
SourceExtractor
+extractSources() : string[]
sends sources
ResultExtractor
+extractResults(reference : string) : string[]
sends results
PluginManager
+retrieveSource() : string
NewsFollower
+checkSources(sources : string[])
NaturalLanguageProcessor
provides source
provides source
uses
provides reference
ArticleReferenceFinder
+findReference(article : Article) : string
provides article
WebParser
+parsePage(source : string) : Article
provides article
ArticleAnalyzer
+analyzeArticle(article : Article) : AnalysisResult
provides analysis
ArticleComparator
-retrieveClaims(analysisResult : AnalysisResult) : ClaimSentence[]
+compareClaims(claims : ClaimSentence[]) : String[]
gathers
analyzes
creates
gets
sends comparison results
Article
-title : Sentence
-body : Sentence[]
+getTitle() : Sentence
+setTitle(title : Sentence) : void
+getBody() : Sentence[]
+setBody(body : Sentence[]) : void
+getAttribute()
+setAttribute(attribute) : void
AnalysisResult
-claims : ClaimSentence[]
+getClaims() : ClaimSentence[]
+setClaims(claims : ClaimSentence[]) : void
ResultSaver
+saveResults(results : string[])
Sentence
-words : Word[]
-isPositive : boolean
-isObjective : boolean
+getWords() : Word[]
+setWords(words : Word[]) : void
+getAttribute()
+setAttribute(attribute) : void
+getIsPositive() : boolean
+setIsPositive(isPositive : boolean) : void
+getIsObjective() : boolean
+setIsObjective(isObjective : boolean) : void
ClaimSentence
compares
Word
-word : String
-root : String
-type : String
+getWord() : String
+setWord(word : String) : void
+getRoot() : String
+setRoot(root : String) : void
+getType() : String
+setType(type : String) : void
+getAttribute()
+setAttribute(attribute) : void
Powered By Visual Paradigm Community Edition

Fig. 4

- **Object Model of Entities**

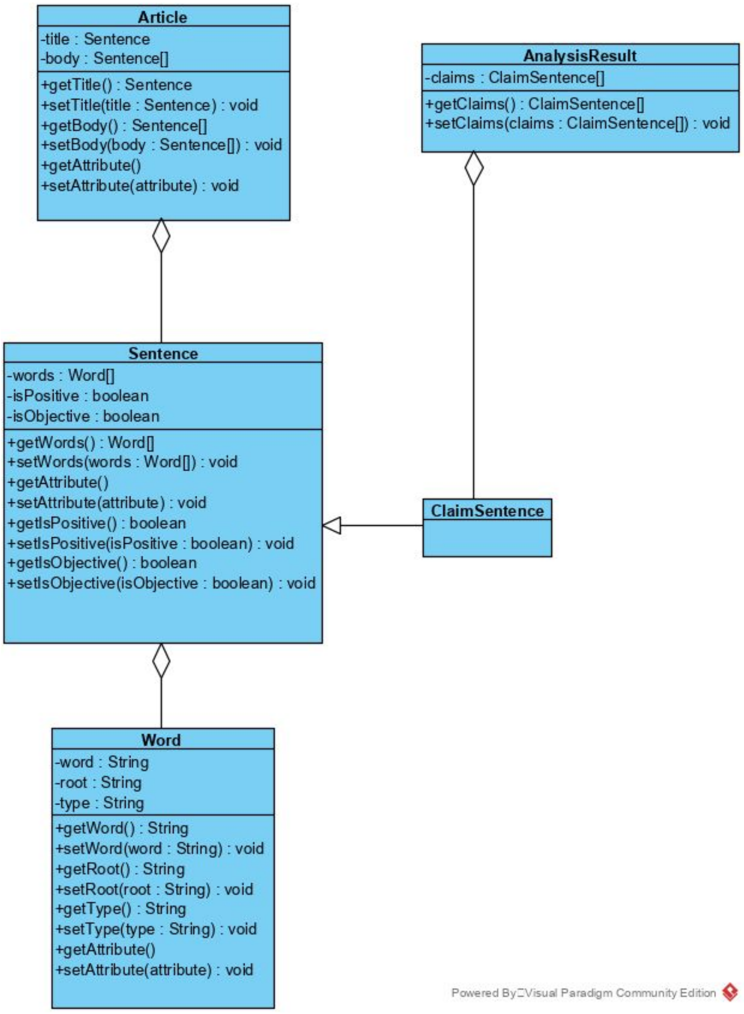

Article
-title : Sentence
-body : Sentence[]
+getTitle() : Sentence
+setTitle(title : Sentence) : void
+getBody() : Sentence[]
+setBody(body : Sentence[]) : void
+getAttribute()
+setAttribute(attribute) : void
AnalysisResult
-claims : ClaimSentence[]
+getClaims() : ClaimSentence[]
+setClaims(claims : ClaimSentence[]) : void
Sentence
-words : Word[]
-isPositive : boolean
-isObjective : boolean
+getWords() : Word[]
+setWords(words : Word[]) : void
+getAttribute()
+setAttribute(attribute) : void
+getIsPositive() : boolean
+setIsPositive(isPositive : boolean) : void
+getIsObjective() : boolean
+setIsObjective(isObjective : boolean) : void
ClaimSentence
Word
-word : String
-root : String
-type : String
+getWord() : String
+setWord(word : String) : void
+getRoot() : String
+setRoot(root : String) : void
+getType() : String
+setType(type : String) : void
+getAttribute()
+setAttribute(attribute) : void
Powered By Visual Paradigm Community Edition

Fig. 5

This model is very similar to the structure of a natural language. Articles consist of Sentences and Sentences consist of Words. The core objective of this model is the Article. The Article objects are created by parsing the news page. Then, they get divided into Sentence objects and the Sentence objects get divided into Word objects. The ClaimSentence objects are obtained after the analysis process. The attributes of ClaimSentence are unknown at this moment and will be determined according to the Natural Language Processing tool we will use. AnalysisResult is the object to store the ClaimSentence objects which are obtained by the analysis.

- **Object Model of Server-Side**

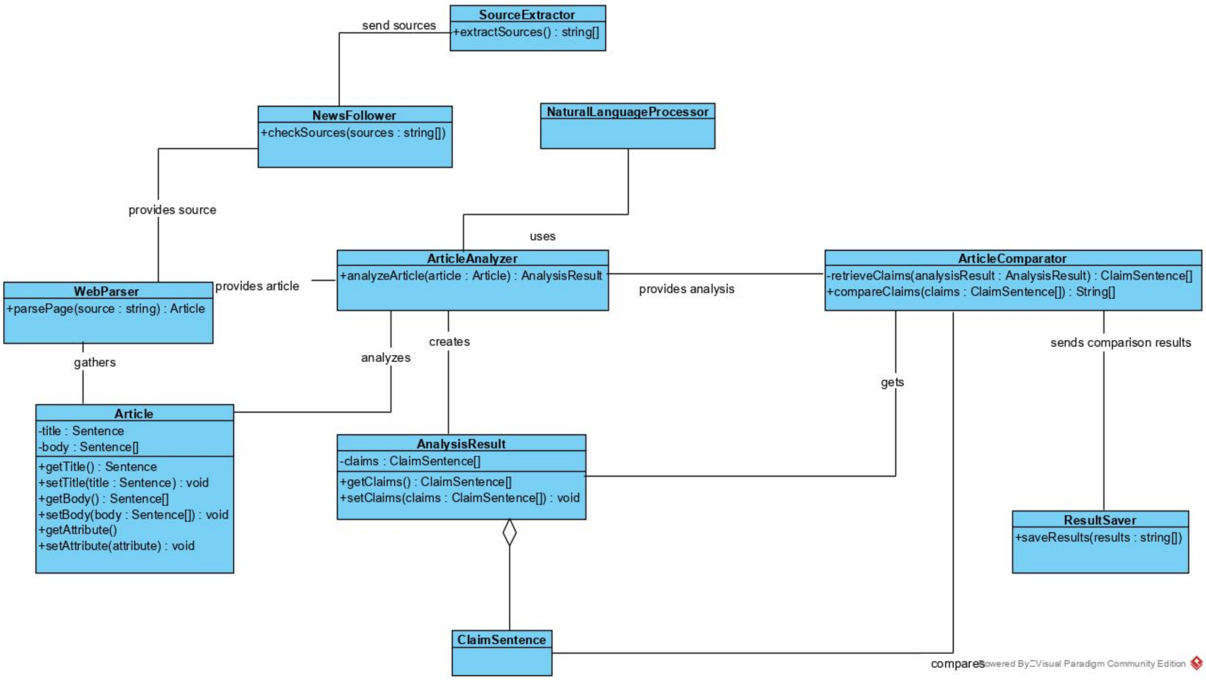

SourceExtractor
+extractSources() : string[]
send sources
NewsFollower
+checkSources(sources : string[])
NaturalLanguageProcessor
provides source
uses
ArticleAnalyzer
+analyzeArticle(article : Article) : AnalysisResult
provides analysis
ArticleComparator
-retrieveClaims(analysisResult : AnalysisResult) : ClaimSentence[]
+compareClaims(claims : ClaimSentence[]) : String[]
WebParser
+parsePage(source : string) : Article
provides article
creates
analyzes
sends comparison results
gathers
gets
Article
-title : Sentence
-body : Sentence[]
+getTitle() : Sentence
+setTitle(title : Sentence) : void
+getBody() : Sentence[]
+setBody(body : Sentence[]) : void
+getAttribute()
+setAttribute(attribute) : void
AnalysisResult
-claims : ClaimSentence[]
+getClaims() : ClaimSentence[]
+setClaims(claims : ClaimSentence[]) : void
ResultSaver
+saveResults(results : string[])
ClaimSentence
compares
Powered By Visual Paradigm Community Edition

Fig. 6

In Figure 6, the objects that are used by the server-side of the program. SourceExtractor is responsible for extracting the web sources where the program checks the news. The NewFollower class consequently checks these sources in order to find out whether there is a newcomer data to any of these sources. If there is, WebParser class parses the web parge the news article located and creates the Article object. ArticleAnalyzer analyzes the Article object using NaturalLanguageProcessor class which is a black box for now since its details are being investigated and creates another object as an output that is the AnalysisResult object which stores the ClaimSentence objects. ArticleComparotor class compares the articles regarding the ClaimSentence instances obtained. At last, ResultSaver saves the results to the database.

- **Object Model of Client-Side**

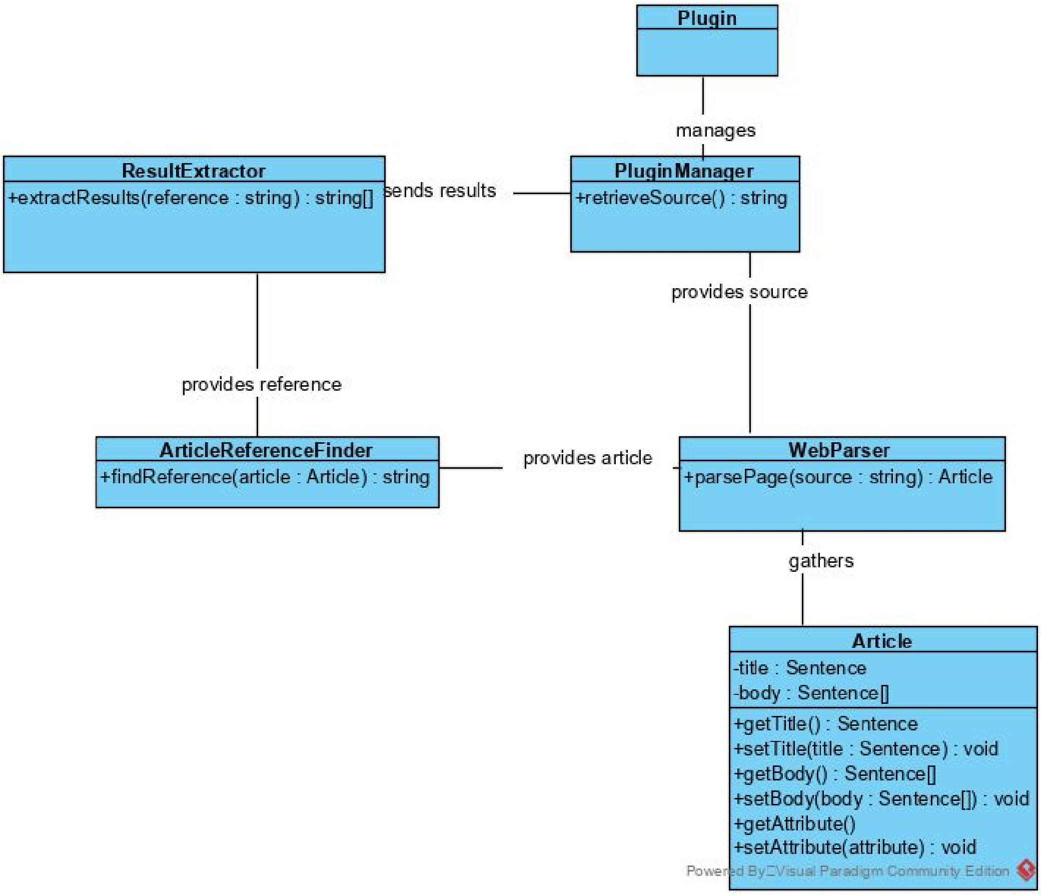

Plugin
manages
ResultExtractor
+extractResults(reference : string) : string[]
sends results
PluginManager
+retrieveSource() : string
provides source
provides reference
ArticleReferenceFinder
+findReference(article : Article) : string
provides article
WebParser
+parsePage(source : string) : Article
gathers
Article
-title : Sentence
-body : Sentence[]
+getTitle() : Sentence
+setTitle(title : Sentence) : void
+getBody() : Sentence[]
+setBody(body : Sentence[]) : void
+getAttribute()
+setAttribute(attribute) : void
Powered By Visual Paradigm Community Edition

Fig. 7

In Figure 7, the objects that are used by the client-side of the program. PluginManager class is responsible for managing the Plugin which interacts with the user. When the user enters a news page, the PluginManager retrieves the source of the web page. WebParser parses the web page and creates an Article instance. ArticleReferenceFinder finds the reference of the corresponding Article instance. ResultExtractor extracts the results that belong to the corresponding article reference from the database. At last, PluginManager makes the Plugin show these results to the user.

3.5.4. Dynamic Models

3.5.4.1. State Diagrams

There are two state diagrams that are the state diagram of the server-side and the client-side. Since they operate separately, they enter different states at different times. Therefore, they have different diagrams from each other.

- **State Diagram of Server-Side**

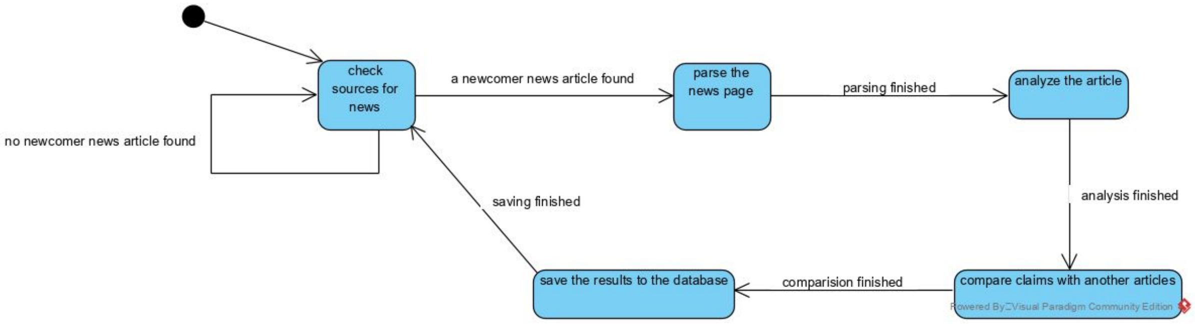

check sources for news
a newcomer news article found
parse the news page
parsing finished
analyze the article
no newcomer news article found
saving finished
analysis finished
save the results to the database
comparision finished
compare claims with another articles
Powered By Visual Paradigm Community Edition

Fig. 8

The model starts with checking sources for news. This state repeats itself until a newcomer news article is found. After it is found, parsing the page of the article starts and ends with an article as the output. Then, analyzing those articles start. After the analysis process is completed, comparing the claims that are found via the analysis with other articles starts. After the results of the comparison are obtained, theses results get saved to the database and the diagrams return to its initial state which is the checking for news state. The diagram always repeats itself as a loop.

- **State Diagram of Client-Side**

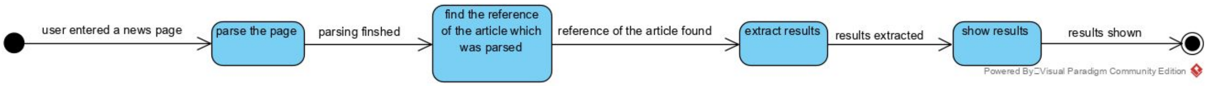

user entered a news page
parse the page
parsing finshed
find the reference of the article which was parsed
reference of the article found
extract results
results extracted
show results
results shown
Powered By Visual Paradigm Community Edition

Fig. 9

This diagram starts with a user to enter a news page. Then, the parsing the page starts and ends with an article as the output such as the server-side. After that, the reference of the article is found and the results of the article with the corresponding reference get extracted. In the end, the results are shown to the user and the reaches its endpoint after the last state ends.

3.5.4.2. Sequence Diagrams

Such as the state diagrams, there two sequence diagrams, as well. The first one is for the server-side and the other one is for the client-side since they operate separately and have different sequences from each other.

- **Sequence Diagram of Server-Side**

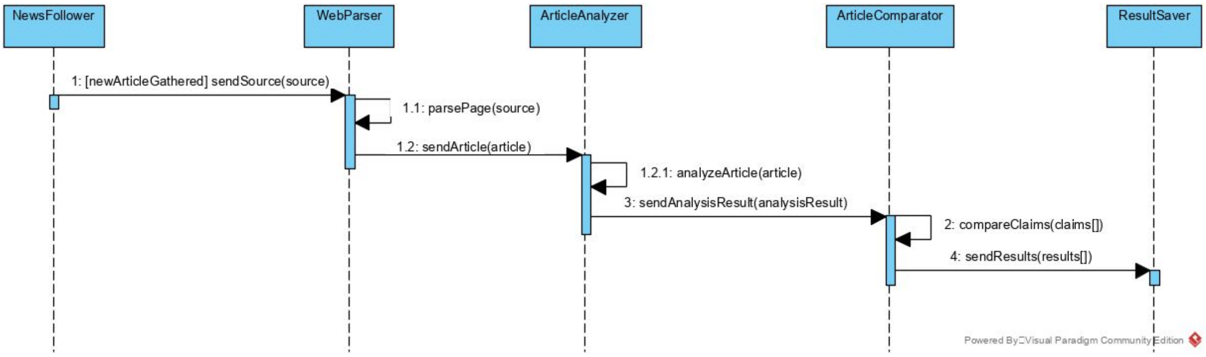

NewsFollower
WebParser
ArticleAnalyzer
ArticleComparator
ResultSaver
1: [newArticleGathered] sendSource(source)
1.1: parsePage(source)
1.2: sendArticle(article)
1.2.1: analyzeArticle(article)
3: sendAnalysisResult(analysisResult)
2: compareClaims(claims[])
4: sendResults(results[])
Powered By Visual Paradigm Community Edition

Fig.10

The sequence of the server-side starts with NewsFollower finding a newcomer news article. It sends the source of that news article to WebParser, then WebParser parses the web page of that source and creates an Article instance. It sends the Article instance to the ArticleAnalyzer and ArticleAnalyzer analyzes the article using NaturalLanguageProcessor. After the analysis process is done, it sends the AnalysisResult to the ArticleComparator. ArticleComparator compares the ClaimSentence instance inside the AnalysisResult and sends the results to the ResultSaver. Finally, ResultSaver saves the results to the database and the sequence of the server-side ends.

- **Sequence Diagram of Client-Side**

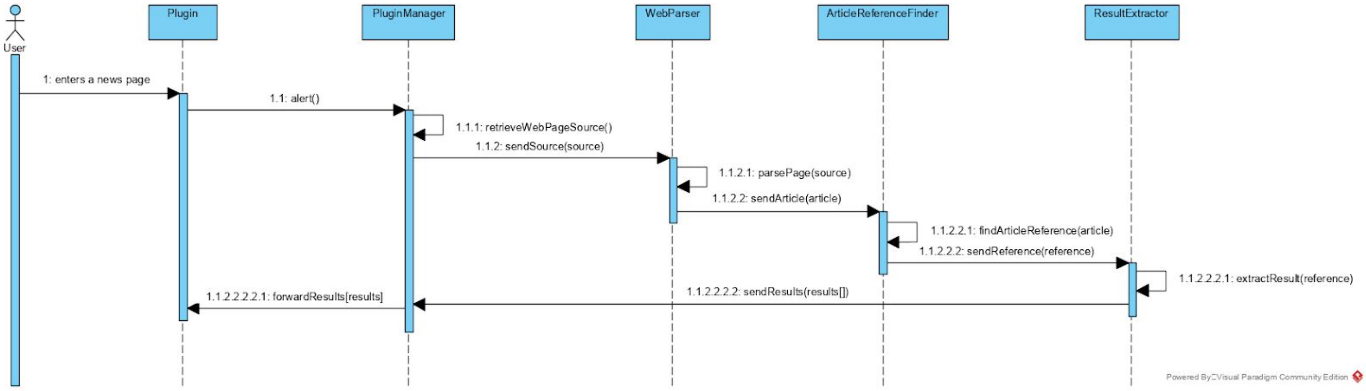
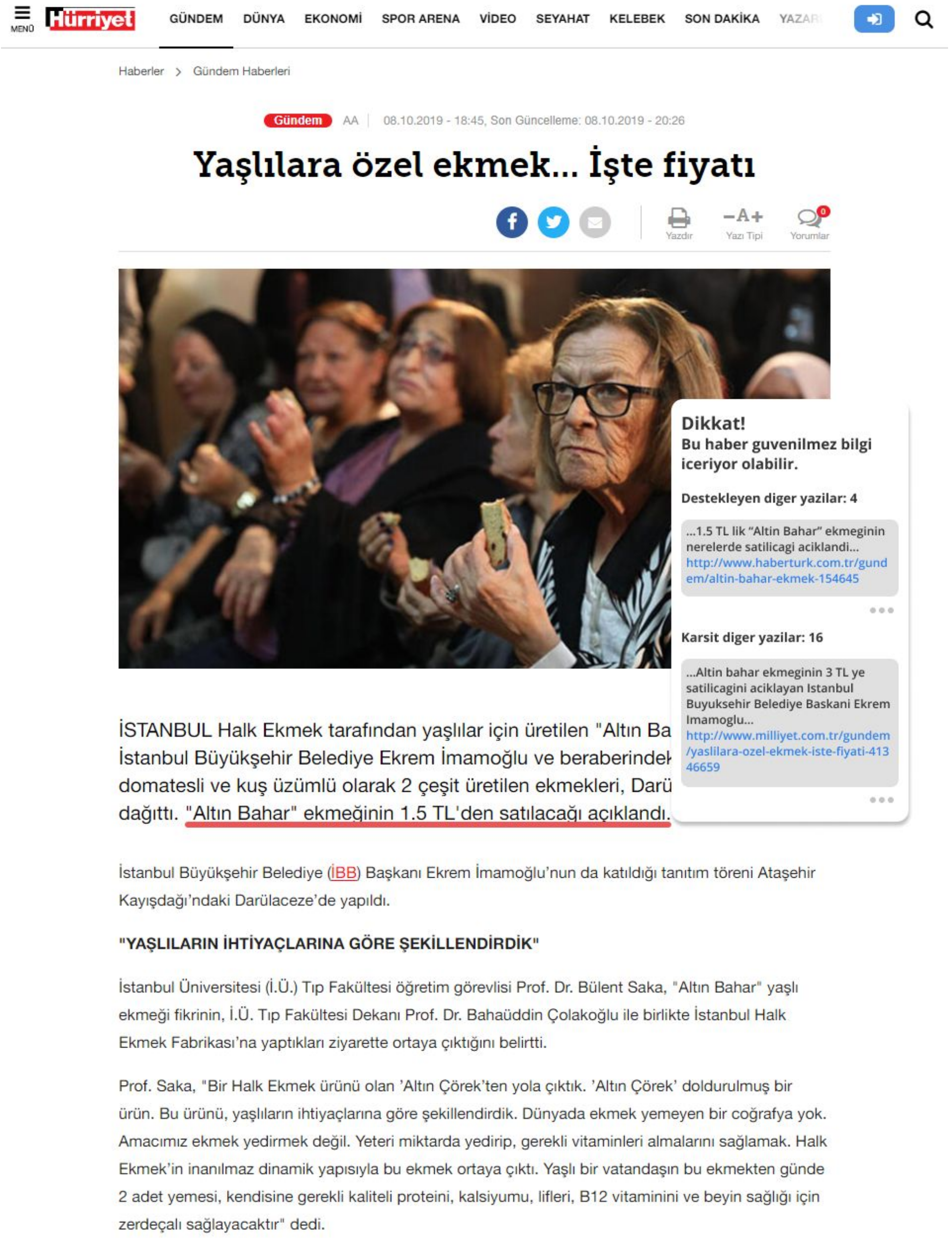

User
Plugin
PluginManager
WebParser
ArticleReferenceFinder
ResultExtractor
1: enters a news page
1.1: alert()
1.1.1: retrieveWebPageSource()
1.1.2: sendSource(source)
1.1.2.1: parsePage(source)
1.1.2.2: sendArticle(article)
1.1.2.2.1: findArticleReference(article)
1.1.2.2.2: sendReference(reference)
1.1.2.2.2.1: extractResult(reference)
1.1.2.2.2.2: sendResults(results[])
1.1.2.2.2.2.1: forwardResults[results]
Powered By□Visual Paradigm Community Edition

Fig. 11

The sequence of client-side starts with the user entering a news page. When the user enters, the Plugin alerts the PluginManager. Then, PluginManager retrieves the source of the news page and sends it to WebParser. WebParser parses the web page and creates an Article instance. ArticleReferenceFinder finds the reference of the corresponding Article and sends the reference to the ResultExtractor. ResultExtractor extracts the results of the Article from the database and sends the results back to PluginManager. At last, PluginManager makes Plugin show these results to the user.

3.5.5. User interface - Navigational Paths and Screen Mock-ups



08.10.2019 - 18:45, Son Güncelleme: 08.10.2019 - 20:26

Yaşlılara özel ekmek... İşte fiyatı

Dikkat!
Bu haber guvenilmez bilgi iceriyor olabilir.

Destekleyen diger yazilar: 4

...1.5 TL lik "Altin Bahar" ekmeginin nerelerde satilicagi aciklandi... http://www.haberturk.com.tr/gundem/altin-bahar-ekmek-154645

Karsit diger yazilar: 16

...Altin bahar ekmeginin 3 TL ye satilicagini aciklayan Istanbul Buyuksehir Belediye Baskani Ekrem Imamoglu... http://www.milliyet.com.tr/gundem/yaslilara-ozel-ekmek-iste-fiyati-41346659

İSTANBUL Halk Ekmek tarafından yaşlılar için üretilen "Altın Ba
İstanbul Büyükşehir Belediye Ekrem İmamoğlu ve beraberindek
domatesli ve kuş üzümlü olarak 2 çeşit üretilen ekmekleri, Darü
dağıttı. "Altın Bahar" ekmeğinin 1.5 TL'den satılacağı açıklandı.

İstanbul Büyükşehir Belediye (İBB) Başkanı Ekrem İmamoğlu'nun da katıldığı tanıtım töreni Ataşehir Kayışdağı'ndaki Darülaceze'de yapıldı.

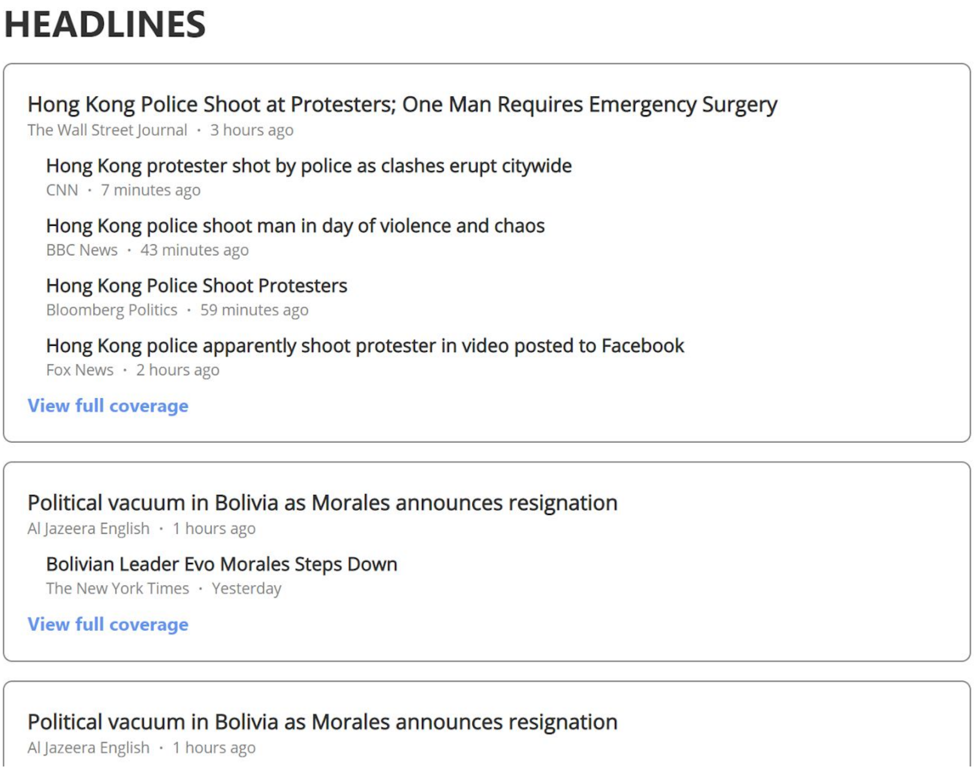
"YAŞLILARIN İHTİYAÇLARINA GÖRE ŞEKİLLENDİRDİK"

İstanbul Üniversitesi (İ.Ü.) Tıp Fakültesi öğretim görevlisi Prof. Dr. Bülent Saka, "Altın Bahar" yaşlı ekmeği fikrinin, İ.Ü. Tıp Fakültesi Dekanı Prof. Dr. Bahaüddin Çolakoğlu ile birlikte İstanbul Halk Ekmek Fabrikası'na yaptıkları ziyarette ortaya çıktığını belirtti.

Prof. Saka, "Bir Halk Ekmek ürünü olan 'Altın Çörek'ten yola çıktık. 'Altın Çörek' doldurulmuş bir ürün. Bu ürünü, yaşlıların ihtiyaçlarına göre şekillendirdik. Dünyada ekmek yemeyen bir coğrafya yok. Amacımız ekmek yedirmek değil. Yeteri miktarda yedirip, gerekli vitaminleri almalarını sağlamak. Halk Ekmek'in inanılmaz dinamik yapısıyla bu ekmek ortaya çıktı. Yaşlı bir vatandaşın bu ekmekten günde 2 adet yemesi, kendisine gerekli kaliteli proteini, kalsiyumu, lifleri, B12 vitaminini ve beyin sağlığı için zerdeçalı sağlayacaktır" dedi.

Fig. 12

A user with the Newspector plugin installed will see a red underline in the news text they are reading when the plugin detects any unreliability within the text. If the user hovers over the underlined text, a pop-up will appear explaining why the text might be unreliable and suggest related articles that might give the user more insight on that particular news and help the user access more sources and decide for themselves if the news they were reading had any problems.



HEADLINES

Hong Kong Police Shoot at Protesters; One Man Requires Emergency Surgery
The Wall Street Journal · 3 hours ago

Hong Kong protester shot by police as clashes erupt citywide
CNN · 7 minutes ago

Hong Kong police shoot man in day of violence and chaos
BBC News · 43 minutes ago

Hong Kong Police Shoot Protesters
Bloomberg Politics · 59 minutes ago

Hong Kong police apparently shoot protester in video posted to Facebook
Fox News · 2 hours ago

View full coverage

Political vacuum in Bolivia as Morales announces resignation
Al Jazeera English · 1 hours ago

Bolivian Leader Evo Morales Steps Down
The New York Times · Yesterday

View full coverage

Political vacuum in Bolivia as Morales announces resignation
Al Jazeera English · 1 hours ago

Fig. 13

The user can also directly access the Newspector website and browse through the news. The news is clustered based on their similarity and shown as headlines on the home page. The user can see the sources of the headlines and the time they were published. The user can click any of the articles on the home page and they will be directed to another page that they can read the article as a whole.

4. Other Analysis Elements

4.1. Risk and Alternatives

Developing a software project comes with its own risks specific to the content of the project. These may be related to different topics such as requirements analysis, implementation, and publishing of the project. As we are developing an application based on Natural Language Processing we are aware of the risks of teaching the human language to machines so that the machines can identify whether the information given in a text is unreliable or not.

- **Requirement Analysis**

One risk is that, in our design process, the requirements for Newspector may not be enough for us to succeed as people would expect new advance features through time from such a system that will be working on the daily news. Therefore, if our requirements are found to be not enough, our alternative is to create a forum where we can connect thousands of news readers around the world, aiming to inform and getting feedback on our design decisions as our target audience is, the people.

- **Implementation**

During the implementation process, there are many risks that we can face. As we are not planning to develop an NLP technology itself from scratch, we will be dependent on the tools and libraries that are already on the market to develop our application. Therefore, this means that we will be mostly manipulating and building our application on the data that is provided us by these tools. The main risk is that the tools and libraries we decide to use may not be sufficient to achieve certain tasks and our alternative is to keep searching on different tools and libraries on the market regularly. Additionally, another way to solve such problems is to develop a new NLP technology for our own purpose of finding and comparing claims among different texts. However, this will not be our first option as it will be highly costly in terms of implementation and testing.

- **Publishing**

The most important risks that we can encounter is in the publishing process of our project. A system like Newspector has the potential to harm many organizations and may be used for manipulating people on specific news about certain people, groups, ideas or movements as well as its positive outcomes. We may become sector enemies with many news publishing websites and organizations so that the amount of advertisement that we can publish on our website may decrease significantly. To solve such a problem, our plan is to avoid labeling news as "Fact" or "Fake" but rather display an unreliability factor that will be based on how different the information provided in a news than the other news published on various websites having the same content. Additionally, while providing information on the reliability of the source websites we will not label these sites as “Reliable” or

"Unreliable" but rather give the information of how many "possibly" unreliable news that the source website provided up to this date and leave the interpretation for the user to avoid any manipulation of ideas of people.

4.2. Project Plan

- **Describing Project Scope, Alternatives, and Feasibility**

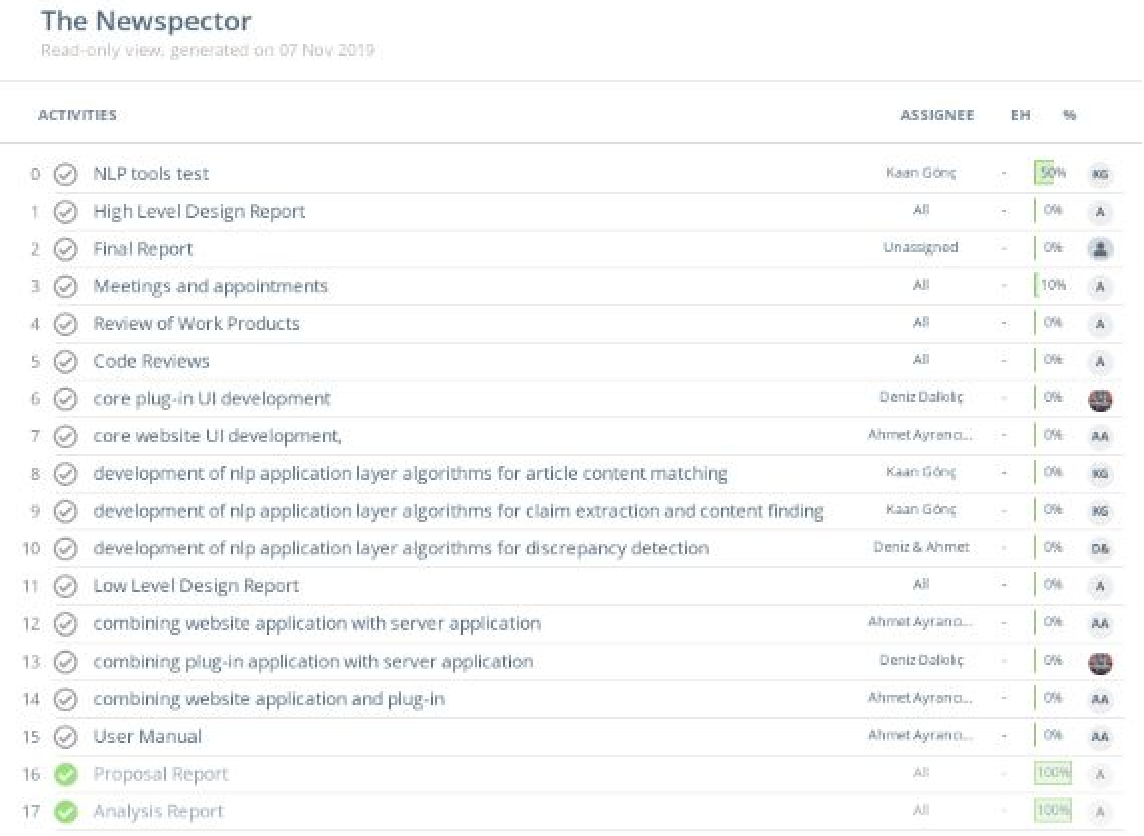
The Newspector is an NLP project which will run its algorithms on news articles published on up to 20 news websites which will be chosen according to the number of daily users. The application should be always doing article analysis and classification as new articles are published on any of these websites. Therefore, it is clear that the scope of our application is broad and the complexity for building an efficient system to accomplish these tasks is high. The Newspector Website and The Newspector Plug-in application are better in terms of feasibility when compared to The Newspector Server Application. Therefore, we need to set our goals, solutions and measurement methods clearly so that we can progress more effectively and know when we reach our goals.

The quantifiable results that we are looking to achieve are:

- The percentage of source websites that we can successfully receive information from should be around 80%.
- The percentage of successfully implemented functional requirements both for the Newspector Website and Newspector Plug-in should be around 85%.
- The percentage of successfully implemented non-functional requirements both for the Newspector Website and Newspector Plug-in should be around 85%.
- The percentage of successfully attracted target users should be around 80%.

- **Dividing the Project into Manageable Tasks**

We are aware that we need to "divide the entire project into manageable tasks and then logically order them to ensure a smooth evolution between tasks." We have a variety of tasks that may be performed in parallel or sequentially. It depends on which tasks produce deliverables needed in other tasks. To arrange our tasks we have prepared high-level Gantt Charts which is a great way of arranging the parallel and sequential tasks in a more organized way. Our Gantt Charts are displayed below:



The Newspector

Read-only view, generated on 07 Nov 2019

		ACTIVITIES	ASSIGNEE	EH	%	
0	✓	NLP tools test	Kaan Gönç	-	50%	KG
1	✓	High Level Design Report	All	-	0%	A
2	✓	Final Report	Unassigned	-	0%	
3	✓	Meetings and appointments	All	-	10%	A
4	✓	Review of Work Products	All	-	0%	A
5	✓	Code Reviews	All	-	0%	A
6	✓	core plug-in UI development	Deniz Dalkılıç	-	0%	
7	✓	core website UI development,	Ahmet Ayranc...	-	0%	AA
8	✓	development of nlp application layer algorithms for article content matching	Kaan Gönç	-	0%	KG
9	✓	development of nlp application layer algorithms for claim extraction and content finding	Kaan Gönç	-	0%	KG
10	✓	development of nlp application layer algorithms for discrepancy detection	Deniz & Ahmet	-	0%	D&
11	✓	Low Level Design Report	All	-	0%	A
12	✓	combining website application with server application	Ahmet Ayranc...	-	0%	AA
13	✓	combining plug-in application with server application	Deniz Dalkılıç	-	0%	
14	✓	combining website application and plug-in	Ahmet Ayranc...	-	0%	AA
15	✓	User Manual	Ahmet Ayranc...	-	0%	AA
16	✓	Proposal Report	All	-	100%	A
17	✓	Analysis Report	All	-	100%	A

Fig. 14

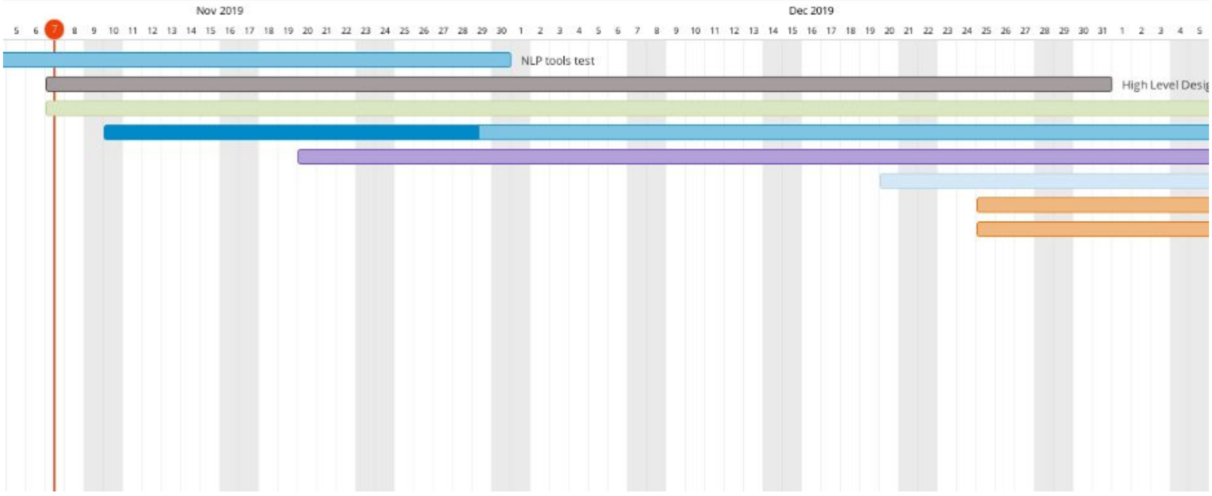


Fig. 15

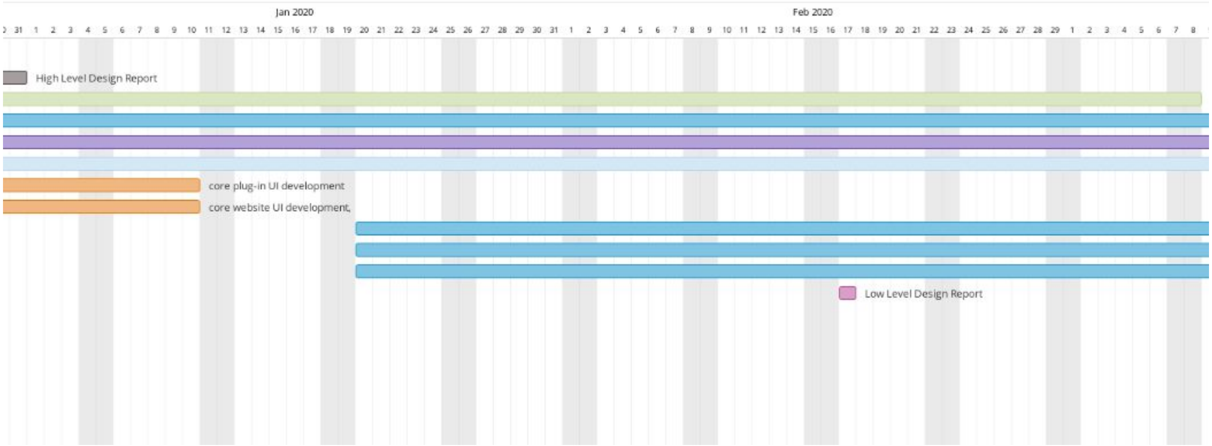


Fig. 16

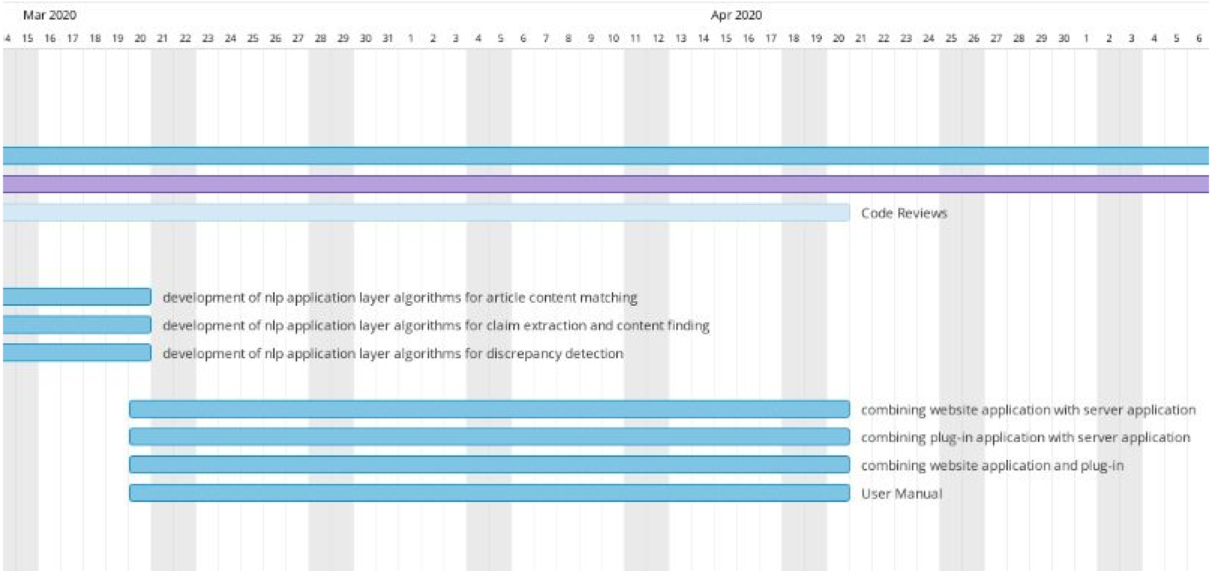


Fig. 17

- **Developing a Communication Plan**

One of the most important aspects of our project process is having an organized and regularly followed communication plan. We should be in touch with possible users of our application as well as communicating regularly with people and organizations who have extensive knowledge of NLP. Our primary contacts are our supervisor Prof. Varol Akman and our innovative expert Mustafa Sakalsız who have great knowledge and experience on the topic. Additionally, we are planning to get in touch with the potential users of our application including our jury members, as they are some of the most important potential users of our application, in this process. Our communication plan consists of:

- Status reports and discussions on problems on the design every week with Varol Akman.
- Discussions with Mustafa Sakalsız on the innovativeness of the design and discussion on possible technologies we can use every 2 or 3 weeks according to our progress.
- Meetings with potential users at every appropriate time.
- Getting in touch with organizations that work on similar NLP technologies to discuss the possible tools that will help us in our project.
- Daily and weekly meetings with the group members to discuss the weekly plan and to inform each other on the current progress.

4.3. Ensuring Proper Team-work

We, as a team of three developers, know that proper teamwork is the key to a successful project as it highly increases the level of productivity and the chances of finding better solutions to problems. For our team to work efficiently, we know that we should keep the quality of communication between group members high and organize our tasks effectively. We have some main bullet points for us to be successful in this process:

- **Making teamwork a priority and rewarding teamwork**

As we are a group of three people, we are aware that if we do not work as a team we won't be able to succeed in such a detailed and big scaled project. Although each individual will have specific parts of the project assigned to them we think that each individual should be aware of what the others are working on and should participate in different discussions in the development process as it will bring diversity and provide better and quick solutions to problems from different perspectives.

- **Clarifying roles, responsibilities, and accountabilities**

In this process, we should identify our roles, responsibilities, and accountabilities in such a manner that we could be able to become leaders in our own responsible fields of the project. This will make the project develop effectively from multiple sides as each team member will have more experience and knowledge on a different part of the project and therefore have the ability to lead others. However, our approach is supporting the idea of each group member having core knowledge of each component in the project so that we can find better solutions to problems by creating a better environment for discussions.

The project can be divided into 3 main parts that the group member can lead. The First one is the Newspector plugin application. The second one is the

Newspector website application. The final part is the back end of both of these applications. A more detailed plan can be seen in the gannt charts in Figures 14 to 17.

- **Setting clear goals**

  We should be well organized during this process. We are planning to do this by setting small and realistic milestones for each of the group members rather than setting bigger and less realistic ones so that the process should carry on much more effectively and interactively as we would move on newer milestones regularly until we reach our goal.

- **Communicating with each other**

  We believe that this point is one of the most important aspects of achieving success in a project. We should keep the quality of our communication between the group members high as it is the key factor for setting goals, solving problems, making design decisions and etc. Additionally, we should keep the variety of our communication methods high so that we could be always active during the project and it will lead us to use different approaches for different problems. By using multiple different communication methods, that are optimal for different goals we can organize the project better and make decisions more clearly. The communication methods that we will use are:

  - Group Meetings
  - Video Conferences
  - Code Reviews
  - Discussions Through Various Communication Networks (Email, Whatsapp, Slack)

4.4. Ethics and Professional Responsibilities

Developing software as a group is a challenging task on its own and continuing this development process without encountering a significant problem in a year-long project makes it even more challenging. To achieve an untroubled development process, communication between group members becomes key. Each member has to act like professionals, treat each other as equal and be aware of the responsibilities that they have to each other. Even though each member has their own schedule and other responsibilities, they have to equally contribute to the project and manage their time to do so.

Newspector will be a browser plugin that will mark the website the user is on. To do so, the plugin has to scrape and read the HTML of the website. Web Scraping can cause legal problems and the website being scraped can sometimes block access if it detects a bot is extracting the data. However, web scraping will only be used for the single page the user is reading and the information gathered from that website will not be used against the interest of the original owner of the data.

Data such as what suggested news articles are clicked or what warnings the user interacts with can be gathered and stored to improve the usability and the effectiveness of the Newspector. However, this information will not be collected without the consent of the users and if the user gives their consent for data gathering, the collected data won't be associated with users directly and will be kept anonymously to increase the privacy of users. Additionally, ant information that will be collected will not be shared with any 3rd parties.

Any software, framework, algorithm, etc. that will be used in the development of the Newspector will be licensed and credited accordingly. Additionally, we will try to use open-source software as much as we can to support and contribute to the open-source community.

Finally, because Newspector aims to increase the awareness of the newsreaders and try to reduce bias and prejudice by suggesting other perspectives and sources for the user to read, there can't be any bias in the program. The developers of the Newspector cannot accept any incentives to favor one newspaper or the other.

4.5. New Knowledge and Learning Strategies

First of all, we need to mention that none of the group members have any experience on the topic of Natural Language Processing. It is totally a new field for us to work on, therefore we can say that our process will be mostly based on learning and applying new techniques and using new tools and technologies to develop our system. We are planning to use the Natural Language Processing tools or libraries that are present on the market as they will provide great help. We need to say that our mission is not to develop a Natural Language Processing technology but to develop a system that will detect the differences between the news having the same content by using the present tools on NLP. The main problem we have is that we have not decided on our target language in The Newspector yet. In order to choose between English and Turkish, we are carrying out research on the NLP tools present on the market for these languages. According to our research, there are many tools that we can use for our purpose. These are:

English

- **NLTK (Natural Language Toolkit for Python)**

  "It implements pretty much any component of NLP you would need, like classification, tokenization, stemming, tagging, parsing, and semantic reasoning. And there's often more than one implementation for each, so you can choose the exact algorithm or methodology you'd like to use. It also supports many languages. However, it represents all data in the form of strings, which is fine for simple constructs but makes it hard to use some advanced functionality." [4]

- **SpaCy for Python**

  “It is probably the main competitor of NLTK. It is faster in most cases, but it only has a single implementation for each NLP component. Also, it represents everything as an object rather than a string, which simplifies the interface for building applications. This also helps it integrate with many other frameworks and data science tools, so you can do more once you have a better understanding of your text data. However, SpaCy doesn't support as many languages as NLTK.” [4]

Turkish

- **Zemberek**

  Zemberek is the most used natural language processing library for Turkish.

- **TRNLTK,**

  TRNLTK is a natural language processing library for Turkish.

  “TRNLTK is built because of the limitations of most used Turkish NLP library Zemberek. TRNLTK and Zemberek teams once tried to work together but it didn’t work because of the time constraints. We’re good friends with Akin brothers (authors of Zemberek) and both teams try to provide a proper solution to Turkish NLP needs.” [5]

  What makes TRNLTK different from Zemberek are:

  - It is highly customizable
  - It is very easy to extend
  - It is less hacky
  - It is much easier to understand the underlying graph
  - It is much easier to maintain
  - Morphologic parser offers more parse results [5]

  Zemberek is better in these subjects:

  - Project is more actively developed
  - Performance is better
  - It offers more tools such as very basic morphologic disambiguator
  - It has a bigger community [5]

- **TRmorph,**

  “TRmorph is a relatively complete morphological analyzer for Turkish. It is implemented using SFST and uses a lexicon based on (but heavily modified) the word list from Zemberek spell checker. The morphological analyzer is distributed under the GPL.” [6]

After gaining experience in these tools and libraries we should follow a strategy that will allow us to be always up to date in terms of the NLP technologies used around the world. As it is mentioned previously, we are currently not sure about which language to build our project on and NLP is a new technology and it develops rapidly, therefore, we should always be aware of the new technologies being developed. We are planning to follow different learning strategies during the project life cycle such as:

- Taking online lessons
- Meetings with Experienced People on NLP
- Participating in Online Forums
- Attending conferences on NLP (if any)
- Reading Books Written on NLP
- Analyzing Example Projects on NLP

5. References

[1] McCarthy, Niall. “Where Concern Is Highest About Fake News On The Internet [Infographic].” *Forbes*, Forbes Magazine, 12 June 2019, https://www.forbes.com/sites/niallmccarthy/2019/06/12/where-concern-is-highest-about-fake-news-on-the-internet-infographic/amp/.

[2] “Oxford Learner's Dictionaries: Find Definitions, Translations, and Grammar Explanations at Oxford Learner's Dictionaries.” *Oxford Learner's Dictionaries | Find Definitions, Translations, and Grammar Explanations at Oxford Learner's Dictionaries*, https://www.oxfordlearnersdictionaries.com/.

[3] Métais Elisabeth, et al. *Natural Language Processing and Information Systems: 24th International Conference on Applications of Natural Language to Information Systems, NLDB 2019, Salford, UK, June 26-28, 2019, Proceedings*. Springer International Publishing, 2019.

[4] D. Barker, “12 open source tools for natural language processing,” Opensource.com. [Online]. Available: https://opensource.com/article/19/3/natural-language-processing-tools. [Accessed: 11-Nov-2019].

[5] “Turkish Natural Language Toolkit,” *Ali Ok*, 02-Oct-2014. [Online]. Available: http://www.aliok.com.tr/projects/2014-10-02-trnltk.html. [Accessed: 11-Nov-2019].

[6] trmorph, 2019. [Online]. Available: http://www.let.rug.nl/~coltekin/trmorph/. [Accessed: 11- Nov- 2019]