ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal
Regular Issue, Vol. 7 N. 4 (2018), 27-34
eISSN: 2255-2863
DOI: http://dx.doi.org/10.14201/ADCAIJ2018742734

# Semantic and Lexical Text Analyzer

Marcos Severt, Álvaro Martín, David Martín, and
Daniel Pérez

marcos_ss@usal.es, id00714312@usal.es, david_martin@usal.es, danipm5@usal.es
Universidad de Salamanca

KEYWORD

*Language Analysis; Lexical and Semantic Analysis*

ABSTRACT

*A multi-agent systems is described that analyzes texts from two points of view: on the one hand in a lexical and on the other in a semantic way. The main purpose of the system is the efficient processing of the inputted text in order to analyzing it, and as a result, outputting it in the right way. That means that after analyzing each phrase of the imputed text, the main agent will delete each wrong phrase. Agents will exchange messages trying to stably which phrase is correct or incorrect. The system will not only remove wrong phrases, it will also make a list with all the removed ones and the reasons that made the main agent discard them so the person that inputted the text can know why those phrases were in a wrong way.*

## 1. Introduction

Nowadays is very common hearing about the use of lexical or semantic correctors in many text processing software. They are used to help the user keeping correct according to the rules of a language both grammar and lexicon. In “Multi-Agent Systems for Natural Language Processing” (Brito *et al.*, 1998), the authors investigate the use of Multi-Agent Systems for natural language processing. They consider a lexical-structural approach and a cognitive-linguistic one and define two types of agents: Reactive agents and cognitive agents, which communicate among themselves in order to increase their knowledge and achieve some goals. In order to solve this, this article proposes a reactive agents architecture where tasks are divided into three groups. Task are solved by four different types of agents, all of them are specialized and their abilities are unique, but they must communicate to solve the problem. This will be achieved by message sending. This system use its own message structure which consists of a java class so we can send it like an object. Message has different attributes that will be explained later in order to make possible that every agent knows what they have to do. System is supplied with one reader agent which will read the inputted text and it will send it’s different phrases making use of the message class to the lexical and semantic analyzer agents. System also has two analyzer agents which will be waiting for a message for determining if the phrase is correct in a semantic or a lexical way. The main goal of the system is reading a text and choose which sentences are correct and which ones are incorrect. The system counts with two reduced dictionaries so we can easily build the software. In order to improve this software it is as easy as improving the dictionaries so they can analyze more phrases to cover a wider language usage. This system will use a group based architecture were cooperative agents work together maximizing productivity in terms of process, time and cost.

A multi-agent arquitecture has been used to develop the proposed system, with different types of specialized agents. This architecture allows efficient parallel text processing. The use of agents and multiagent technology allows to carry out smart management of a complex system, coordinating the different subsystem that compose it, as well as integrating objectives proper of each subsystem with a common objective. Considering each subsystem with a local decision capability, management issues can be approached with a cooperative and coordinated/negotiated perspective among agents in order to achieve an efficient way to solve a problem. After reviewing some projects in this fiels, the proposed architecture is proposed and the system is evaluated, finally conclusions ara presented.

## 2. RelatedWork

Nowadays text files analysis is widely used in many software Applications. That is why many researchers are searching for more eficient methods of analysis. One of the primary goals intext-comprehension research is to understand what factors influence a reader's ability to extract and retain information from textual material. Latent semantic analysis (LSA) is a statistical model of word usage thtat permits comparison of semantic similarity between pieces of textual information. The LSA method can also be used for a very different type of analysis used in text comprehension, the measurement of coherence (Foltz 1996).

Sentiment Analysis (SA) is an ongoing field of research. It is known as Opinion Mining or emotion AI, and it refers to the use of natural language processing, text analysis, computational linguistics and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment Analysis can be considered a classification process. There are three main classification levels in SA: document-level, sentence-level, and aspect-level (Medhat *et al.*, 2014). SA uses a lot of information so one of the most important subtasks of SA is subjectivity detection. In other words, it must remove the factual or neutral comments that lack sentiment (Chaturvedi *et al.*, 2018). SA semantic approach is characterized by a efficient use of lexical terms dictionaries. SA Semantic Systems process the text with the appropriate elimination of stop words and linguistic normalization by stemming, and then they check the appearance in terms of lexicon to assign the polarity value of the text by adding the polarity value of the terms (Román, 2015).

Agent-Oriented Programming (AOP) is a relatively new software paradigm that brings concepts from the theories of artificial intelligence into mainstream real of distributed systems. AOP essentially models and application as collection of components called agents that are characterized by, among other things, autonomy, proactivity and an ability to communicate (Bellifemine *et al.*, 2007). Agent-based computing has often been suggested as a promising technique for problem domains that are distributed, complex and heterogeneous. In particular, a number of agent-based approaches have been proposed to solve different types of resource allocation problems (Davidsson *et al.*, 2007). The investigation of methodologies for analysis and design of Multi-agent Systems (MAS) is still in embryonic state. The existing methodologies for developing MAS are not exactly new; generally they are extensions from object-oriented methodologies or knowledge engineering methodologies, given their close relationship (Aguilar *et al.*, 2007). Multi-agent systems are being used in an increasingly wide variety of applications, ranging from comparatively small systems for personal assistance to open, complex, mission-critical systems for industrial applications (Bellifemine *et al.,* 2007).

Nowadays social network have changed people`s lifestyles and the way people communicate and relate. One main application of multi-agent Systems is text mining for recommended relationship System in a business and employment-oriented social network. Text mining is the process of deriving high- quality information from text.

High quality information is typically derived through the devising of patterns and trends through means such as statical pattern learning. In order to make this possible this System uses distributed multi-agent systems which have become very sophisticated in the last years, with a rising potential to handle large volumes of data and coordinate the operations of many organizations (Chamoso *et al.*, 2018)

Clickbait (content whose main purpose is to attract attention and encourage visitors to click on a link to a particular web page) is one of the biggest problems of our time. That is why multi-agent systems can solve problems like this. Several researchers have tried to detect clickbait by applying different techniques. CBR

(Case Based Reasoning) is one of these techniques which involves solving new problems based on the solutions of similar past problems. CBR starts with a set of cases or training examples; it forms generalizations of these examples, albeit implicit ones, by identifying commonalities between a retrieved case and the target problem. Case-based reasoning is a prominent type of analogy solution making (López Sánchez *et al.*, 2017)

Since almost any laboratory involved in agent research has developed its own agent architecture, agent architectures are nearly as numerous as agent systems in general. Traditionally, they have been classified according to their roots in either logic-based Artificial Intelligence or behavior-based Artificial Intelligence. The first class is called deliberative, the second reactive agent architecture (Hannebauer, 2003). This system uses a reactive agent architecture to solve all it's tasks.

Text analysis is not only used for natural language. A compiler has several components, being three of them Lexical Analysis, Syntax Analysis and Semantic Analysis. Both first and third are used in "Multi- Agent Systems for Natural Language Processing" (Brito *et al.*, 1998) and this system uses both the first and the third ones. In order to make this system we have chosen JADE (Java Agent Development Framework) which is very popular because of its open source, simplicity and compliant with the FIPA specifications (Bala- chandran 2008). There are so many methodologies for the development of multi- agent Systems using JADE platform. We have made use of a methodology which focuses on the key issues in the analysis and design of multy agent Systems. The analysis phases is generic, while the design phase specifically focuses on the constructs provided byt FIPA-compliant JADE platform, as it is said in (Nikraz *et al.*, 2006).

# 3. Text Sentence Analyzer (TSA)

The objective of the system is to process an inputted text and determine which phrases are correct and which are not. It will make use of the different agents and dictionaries that will be explained below. This system has a reactive architecture which means that the agents have a simple internal representation of the world and a behavior-based paradigm. Their different functions will be made as long as they receive a message from other agent which is considered as an external stimulus. They know how to act depending on the received message so they can not act on their own.

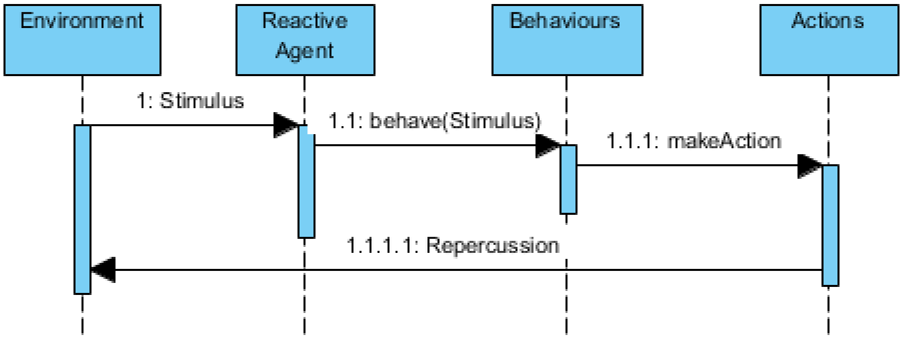

*Figure 1: Reactive Agent Description*

Agents Organization means having a collection of objective-related roles. Agents have interrelation which allows establishing any order within them. In this case the order used is a group based multi-agent organization where agents work together in the achievement of a common objective. In order for cooperation to exists tasks are distributed based on agent's specialization. This system has the following tasks:

- **Reading the inputted text:** This task is made by the reader agent which will read each phrase of the text and will send it to the analyzer agents.
- **Controlling redundancies:** This task is made by the reader agent which will store each phrase after reading it so it can know any other time if those phrase are correct.

- **Lexical analysis:** This task is made by the lexical analyzer agent which will read each message from the reader agent and then it will analyze the message's phrase.
- **Semanticanalysis:**This task is made by the semantic analyzer agent which will read each message from the reader agent and then it will analyze the message's phrase.
- **Phrase verification:** This task is made by the supervisor agent which will read each message from both of the analyzer agents and will send a message to the reader agent in order to making him know if the phrase is correct or incorrect.

The following class diagram is used to solve this issue:

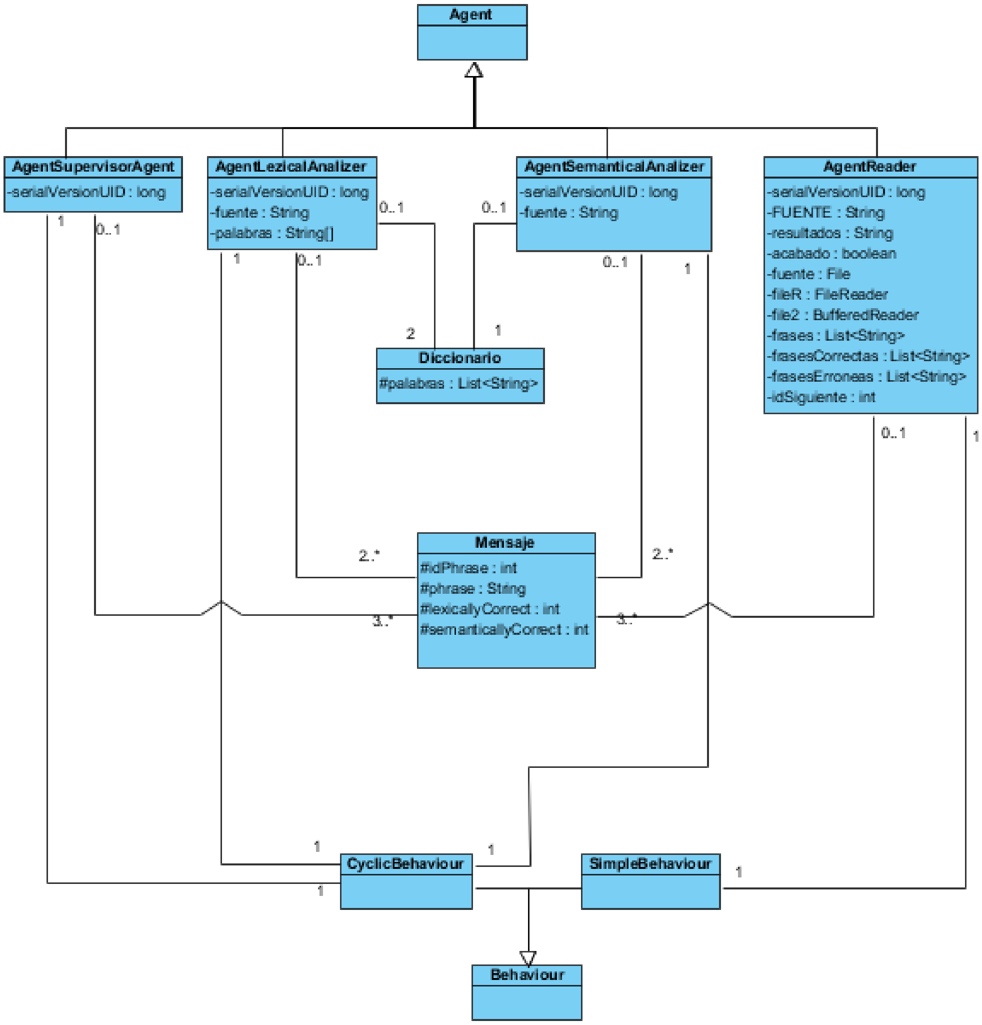

*Figure 2: Class Diagram*

This system will have the following group-based architecture system in order to achieve all tasks:

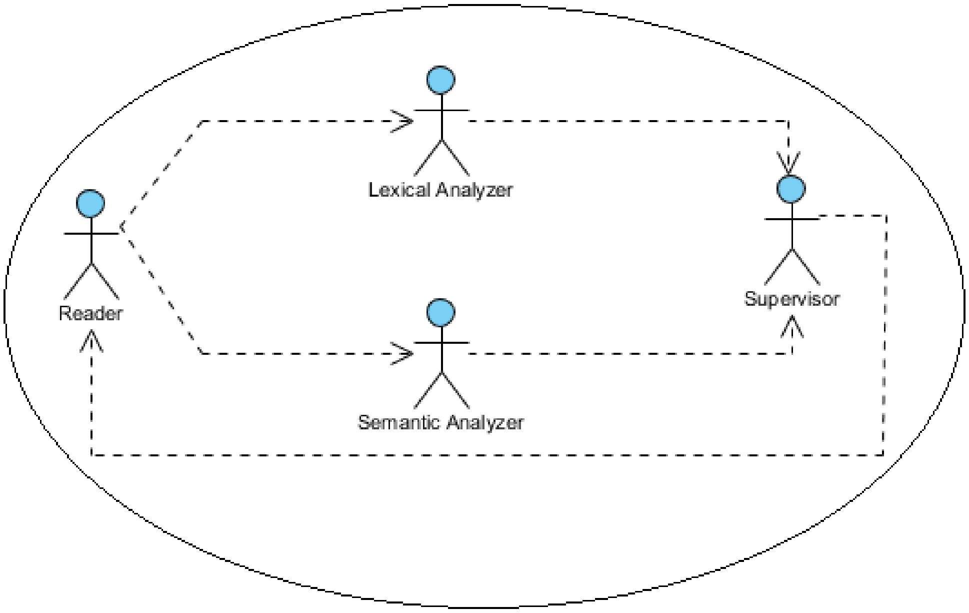

*Figure 3: Group Architecture*

An agent is an encapsulated computational system inside an environment who acts in an autonomous and flexible way so it can reach its goals. It has some properties such as autonomy, sociability and reactivity. They need to communicate between themselves to achieve their goal. While the initial model defines reactive (stimulus-response agents) and cognitive (reasoning agents) types (Brito *et al.*, 1998), our system defines four reactive agents:

- **Agent Reader:** One of the reactive agents, who is on charge of reading the chosen text. The Reader is an intelligent agent since it cooperates with other agents by sending them the read text and has kind of a learning capability, being able to detect repeated sentences. When the text is read, it is divided in simple sentences the Reader stores into its database. Each sentence has an unique ID number when read. Next, those sentences are sent to the Lexical Analyzer and the Semantic Analyzer. Then the Reader will receive messages from the Supervisor with information about the sentences. Its beliefs are several text files with information about the text and the results. The Reader has also three collections with the initial sentences and the analyzed ones. Its desire is to get every sentence analyzed. At this point, a boolean variable will become true, finishing the execution. Finally, it performs a set of actions to achieve its goal. First, it reads the whole text. Then, it sends the collection of sentences to the analyzers in order to receive a corrected one. Finally, the Reader prints the results on screen.
- **Supervisor Agent:** Supervisor is the other reactive agent which control which sentences are correct and which ones not. It is an cooperative agent but it depends on the phrases sent by the analyzers to actually do something. It is receiving messages from the Lexical and Semantic Analyzers with information about the sentences.

  Every sentence has an ID number. This allows the Supervisor to match the information it gets from the analyzers and store the sentences in its own database. When the analysis is finished, the user can ask the system to print sentences on screen. Supervisor is the agent responsible of this. It can either print Inputted Text (The read text is showed on screen), Correct Sentences (The corrected text is showed on screen) and Incorrect Sentences (Every incorrect sentence is showed on screen as well as its mistakes). The Supervisor's desire is to receive the results from the Analyzers and then send them back to the Reader agent. Its plan consists of exchanging messages between the rest of the agents until the whole text is analyzed.

- **Lexical Analyzer:** Lexical Analyzer will receive messages from the reader and it will look up each received word in the dictionary. If all the words were found it will send a message to the supervisor agent. In order to make this possible this agent will load its dictionary in its setup so when it starts receiving messages it is ready to analyze them. This Analyzer has two main beliefs. A dictionary, used in the analysis comparation, and an array of words where it stores the sentences. Its desire is to lexically analyze all the sentences the Reader sends. To achieve this goal, it recieves a group of sentences from the Reader and then compares them with its dictionary. Finally, it sends the results to the Supervisor.
- **Semantic Analyzer:** Semantic Analyzer will receive messages from the reader and it will look up each received phrase in the dictionary. If all the words were found it will send a message to the supervisor agent. In order to make this possible this agent will load its dictionary in its setup so when it starts receiving messages it is ready to analyze them. Both lexical and semantic analyzers make use of the dictionary java class which will be explained later. The main belief of the Semantical Analyzer is its dictionary. The agent uses it in order to achieve its goal: getting every received sentence semantically analyzed. It performs several actions, such as receiving the text, comparing the sentences using the dictionary and sending them to the Supervisor Agent.

Communication between agents allows them to interchange information about the analyzed phrases and coordinate in order to making possible the proper function of the system avoiding all kind of conflict. This system uses global coordination which means that the multi-agent system provides coordination and the means of realistically optimizing the processing efficiency. Communication in the system consists of message passing functions. Messages will have the following structure:

*Table1: Message Structure*

| **Agents** | FROM READER AGENT TO ANALYZER AGENT | |
|---|---|---|
| **Message Structure** | PHRASE ID | PHRASE FROM THE TEXT |

| **Agents** | FROM SUPERVISOR AGENT TO READER AGENT | |
|---|---|---|
| **Message Structure** | PHRASE ID | IS PHRASE OK |

| **Agents** | FROM LEXICAL ANALYZER AGENT TO SUPERVISOR AGENT | | |
|---|---|---|---|
| **Message Structure** | PHRASE ID | PHRASE FROM THE TEXT | LEXICALLY OK |

| **Agents** | FROM SEMANTIC ANALYZER AGENT TO SUPERVISOR AGENT | | |
|---|---|---|---|
| **Message Structure** | PHRASE ID | PHRASE FROM THE TEXT | SEMANTICALLY OK |

This message structure allows the following phrase processing sequence:

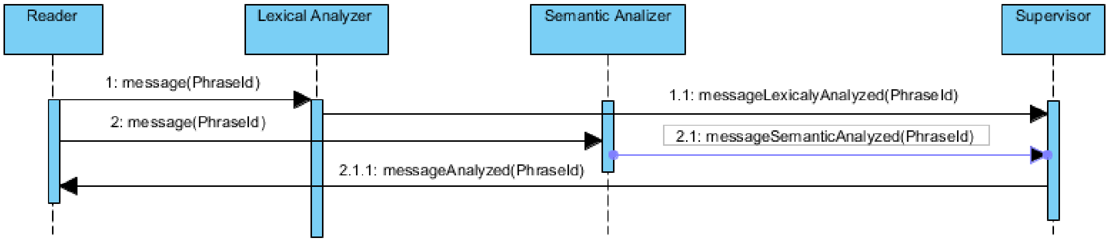
Reader
Lexical Analyzer
Semantic Analizer
Supervisor
1: message(PhraseId)
1.1: messageLexicalyAnalyzed(PhraseId)
2: message(PhraseId)
2.1: messageSemanticAnalyzed(PhraseId)
2.1.1: messageAnalyzed(PhraseId)

*Figure 4: Communication Diagram*

Both lexical agent and semantic agent, use a dictionary. The lexical dictionary is composed by words, while the semantic dictionary is formed by short sentences. When a phrase is sent to the lexical agent, analyzes word by word and checks if every of them is included in its dictionary. In the case of the semantic agent, when a phrase arrives, it is analyzed as a whole sentence, in other words, the semantic agent looks up to the exact sentence in the dictionary. In case every Word of the sentence is in the lexical dictionary, and the sentence itself is in the semantic dictionary, that phrase is validated. This system make use of its own dictionary class which provides methods that allow the user to create a new dictionary or to find a word/phrase in order to analyze each phrase.

## 4. Results and Conclusion

For test realization, it has been chosen a particular case for lexic-semantic translators. The lexical analyzer owns a 8500 word dictionary in Spanish, and 500 names. The semantic analyzer has 4000 sentences. A dictionary with wordly renown phrases has been selected to fulfil the tasks, given that this small case is enough for the test. Whichever the introduced test is, it is segmented in paragraphs and eventually in sentences, analyzing each sentence one by one, and deciding if tis correct or not, and the reasons of it. Due to the small scale of the test, a sequential processing realization seemed suitable. However, the eficinecy could be improved by increasing the number of agents, or analyzing more than one word at the time. The obtained results show that the usage of a wider dictionary would not affect negatively in the efficiency, as well as he methods for reading or writing.

The input file has been modified for testing that whatever the test is introduced, the system would be able to analyze completely and rightly each introduced sentence. For avoiding errors, the system erases blank spaces at the beginning and the end to accomplish the test accurately. The result given by the analysis can be shown in a table, or presented in its own file. A choosing system could be implemented at this part, where the user would like to save the results, or adding another formats like ods or csv, not only odt.

This system version shown by this article is a reduced one. In order to improve effectiveness, dictionaries should embrace all language words. In case of the semantic dictionary, it should also embrace all correlations between nouns and verbs. Multiple solutions could be reached due to the freedom provided. In this case, 4 agents were seemed suitable in order to solve the problem, each of them responsible of a single task. However, it is not denied that a more effective approach could have been reached. This idea can be used as the foundation of similar system-based ideas. For instance, the system structure would benefit a system whose objective lies in building a coherent text narrating events happening during a period of time. All algorithm used in this system can be improved to achieve a better efficiency.

This system could be improved by adding self learning for both analyzer agents so they can improve their effectiveness. In this case Reader Agent sends each phrase and waits for each analysis, it could be improved by sending all the phrases and then waiting for the analysis making use of phraseID argument.

## 5. References

Aguilar, J., Cerrada, M., and Hidrobo, F., 2007. A Methodology to Specify Multiagent Systems. In Agent and Multi-Agent Systems: Technologies and Applications, pages 92-93. Springer.

Balachandran, B. M., 2008. Developing Intelligent Agent Application with JADE and JESS. In Know ledge-Based Intelligent Information and Engineering System, pages 236-238. Springer.

Bellifemine, F., Caire, G., and Greenwood, D., 2007. Developing Multi-Agent Systems with JADE. Davidsson, P., Persson, J. A., and Holmgren, J., 2007. On the Integration of Agent-Based and Mathematical Optimization Techniques. In Agent and Multi-Agent Systems: Technologies and Applicat ions. Pages 1-20.

Carvalho, A. M., da Silva, D., Tavares, J. L., and Strube, V. L., 1998. Multi-Agent Systems for Natural Language Processing. Pages 1-5.

Chamoso, P., Rivas, A., Rodríguez, S., Bajo, J., 2018. Relationship recommender System in a business and employment-oriented social network. In Information Sciences, pages 204-219. Elsevier.

Chaturvedi, I., Cambria, E., Welsch, E., R., Herrera, F., 2018. Distinguishing between facts and opinions for sentiment analysis: Survey and challenges. In Information Fusion, pages 65-77. Elsevier.

Foltz, W., P., 1996. Latent semàntic analysis for text-based research. In Analysis of semàntic and clinical data, pages 197-201. Springer.

Hannebauer. M., 2003. Compostable BDI Agents. In Autonomous Dynamic Reconfiguration in Multi-Agent Systems, pages 138-142. Springer.

López Sánchez, D., Revuelta, H., J., González, A., A.,Corchado, R., J. M., 29 December 2017. Hybridizing Metric Learning and Case-Based Reasoning for adaptable clickbait detection. In Applied Intelligence. In press. Springer.

Medhat, W., Hassan, A., Korashy, H., 2014. Sentiment analysis algorithms and Applications: A survey. In Ain Shams Engineering Journal, Volume 5, Issue 4, pages 1093-1113.

Nikraz, M., Caire, G. and Bahri, P.A., 2006. A Methodology for the development of multi-agent System using the JADE platform. In International Journal of Computer Systems Science & Engineering, pages 99-116.

Roman, V., J., 2015. Introducción al anàlisis de sentimientos (mineria de opiniones). Pages 1-3.