A Multi-Agent System for guiding users in on-line social environments

G. Aguado[a,*], V. Julian[a], A. Garcia-Fornes[a], A. Espinosa[a]

[a] *Valencian Research Institute for Artificial Intelligence (VRAIn)*
Universitat Politècnica de València
Camino de Vera s/n, Valencia, Spain

Abstract

The present work is a study of the detection of negative affective or emotional states, the high-stress levels that people have using social network sites (SNSs), and the effect that this negative state or stress level has on the repercussions of posted messages. We aim to discover to what extent a user that has a state detected as negative by an analyzer (Sentiment analyzer and Stress analyzer) can affect other users and generate negative repercussions, and also determine whether it is more suitable to predict a future negative situation using different analyzers. We propose two different methods for creating a combined model of sentiment and stress, and we use them in our experimentation to discern which one is more suitable for predicting future negative situations that could arise from the interaction between users, and in what context. Additionally, we designed a Multi-Agent System (MAS) that integrates the analyzers to protect or advise users on a SNS. We have conducted this study to help build future systems that prevent negative situations where a user that has a negative state creates a repercussion in the SNS. This can help users avoid getting into a bad mood or help avoid privacy issues (e.g. a user that has a negative state posting information that the user does not really want to post).

Keywords: Multi-Agent System, Social Networks, sentiment analysis, stress

☆Fully documented templates are available in the elsarticle package on CTAN.

*Corresponding author

Email addresses: `guiagsar@dsic.upv.es` (G. Aguado), `vinglada@dsic.upv.es` (V. Julian), `agarcia@dsic.upv.es` (A. Garcia-Fornes), `aespinos@dsic.upv.es` (A. Espinosa)

analysis

1. Introduction

In our current society, people are immersed in an environment of on-line applications, of which social networks or Social Network Sites (SNSs) are the most important and most ubiquitous. One question that arises from this social interaction between users is whether or not users are safe. In [1], risks and negative outcomes arising from the interaction between users in a SNS have been reviewed. In [2], and [3], important risk factors are reviewed. Some of the most important risk factors are content risks, which are the risks of receiving inappropriate content, which can be varied (e.g. pornography, violence, and racism). Other important risks are contact risks, which are the risks that arise from meeting strangers and interacting with them. This can lead to cyber-harassment, privacy issues, and potentially harmful chat contacts. There are also commercial risks, which involve people receiving spam or getting asked for personal information, which also can lead to spam or aggressive marketing. It is also important to note that teenagers face several risks on SNSs and have characteristics that make them more vulnerable to them [4].

The decision making of users of on-line social platforms determines the way they interact, and an unfortunate choice may lead to incurring the risks mentioned above. For example publishing a post about private aspects of the user that attracts sexual predators, thus falling into contact risks, or publishing a post about the violence that could attract unwanted content about violence, incurring content risk. Decision making has been shown to be affected by the emotional state of the person making the decision. The effect of incidental moods, discrete emotions, integral affect, and regret on decision making have been reviewed in [5]. By incidental moods and discrete emotions, we mean affective states that are not directly linked with the task at hand and that can arise from other sources (e.g., emotion arising from making decisions, thinking of someone, talking to someone that is not directly linked to the task being performed). On the other hand, integral affect is generated from the task being worked on. Finally, regret is a negative and conscious emotional reaction to self decision-making. In the review, the authors show that incidental moods affect decision making by altering people's perception, and also that discrete emotions, integral affect, and regret affect decision making (regret acts as anticipated regret, thinking of the negative outcome before it actually happens). Moreover, stress has also been observed to be associated with a specific emotional state (high arousal and negative valence) and has been used in [6] to construct an adaptation (TensiStrength) of the sentiment strength detection software called SentiStrength [7] for detecting stress and relaxation magnitude in texts. Taking into account the previous, stress levels may be suitable for building a system that analyzes the state of the users along with sentiment values.

As stated above, the emotional state of the user can influence decision making. This can lead to future problems in a social on-line environment and may also make users fall into risks derived from their interaction. As an example of decision making resulting in a negative outcome, in [8], the authors show that when a user publishes a post, it can lead to regret and have negative consequences. Thus, it would be desirable for a system to be able to detect this sentimental state of the user and to react to it by trying to advise or protect him or her from possible future negative outcomes that could arise from his or her behavior.

Following the general idea of a system that analyzes the emotional state and stress levels of a user when he or she is interacting on on-line sites, in this work, we present a Multi-Agent System (MAS) for assessing guiding users in SNS by performing sentiment analysis, stress level analysis, and a combined analysis on user posts, and potentially giving them feedback if necessary. The system is built as a MAS to allow the tasks of different analyses to be performed separately and also, to allow the system to start processing new user input while still analyzing a previous one. This is possible due to the pipeline of agents that is built into the architecture, which is shown and discussed in Section 3. Different agents perform distinct analyses, and there are also other agents for the interaction with the users in the on-line social environment and for advising and retaining/retrieving data. This system has been integrated into a SNS to guide the users in their experience through the social environment by advising them when they are going to post messages, analyzing the text of the message with the different analyzers, and warning the user (or not), depending on the results of the analyzers in order to prevent a possible bad outcome (e.g., triggering an argument with other people or publishing content that the user does not really want to make public because of cognitive distortions). This MAS is a modification of a previous prototype presented in [9], where the analyzers were built using a Bayesian classifier. In the current version, we built the analyzers using feed-forward Artificial Neural Networks (ANN), which have been coded using the Tensorflow[1] and Keras[2] libraries with the programming language Python. We used ANNs to improve the classification accuracy and performance of the system since machine learning techniques have been used for aspect-based sentiment analysis achieving state-of-the-art accuracies [10]. In [9], we conducted a set of experiments with data from Twitter.com to determine which analysis was able to detect a state of the users that was propagated the most to the replies of the messages. We used the most present value in the replies as a metric of propagation so the analyzers detecting a state of the user that has high propagation would be more useful for detecting messages that generate problems in the future in a SNS. Since none of the analyzers showed a significant difference against the others, in this paper, we present new experiments with the new analyzers, using a new version of the combined analysis, and also show that one of the versions of combined analysis achieved to perform significantly better than the others.

The contribution of the present work is twofold. On the one hand, we constructed a new version of our MAS introducing new analyzers using ANN, and we used our MAS in experiments in a laboratory with a SNS called Pesedia [11] that was used by a set of children, whose ages were between twelve and fifteen years old, and we were able to draw conclusions about how the proposed MAS works in a real-life environment. On the other hand, we extracted conclusions from experiments performed with data from Twitter.com to determine which analyzer predicts a state of the user that propagates more to the replies of the messages.

Regarding the advantages of our proposal comparing it to the state-of-the-art works, our proposed approach leverages the use of both MAS technologies and ANN to try to accomplish the task of prevention of potential issues, negative outcomes or propagation of negative sentiment polarity or high-stress levels on an on-line social environment, using for this purpose two sources of data, which are the sentiment polarity and stress levels of users interacting with the social environment, and proposing a combined analysis with two modalities. To the best of our knowledge, the state-of-the-art works only use one of those input data sources to prevent negative outcomes in SNSs. We also performed experiments to discover which of the analyses, including the combined modalities, should be used to be more informative in the system and in which cases. This is not the case on the current state-of-the-art works. One of the modalities of the combined analysis shown in the experiments performed in the current work that it can detect a state of the user that significantly propagates more in the network than the other analyzers, which is an advantage when creating a system that warns users based on the analyzed state on their messages. Related to the limitations of the current approach, as we created a system to be used integrated

[1] https://www.tensorflow.org
[2] https://keras.io

into a SNS for people of young age, we used a dataset made from texts written and labeled by people aged between twelve and fifteen years old for training the machine learning models. Using more datasets made from people of varied ages for creating different models and testing them could improve the performance of the system. Nevertheless, our experiments have shown that the system is able to perform as intended, as will be shown in sections 4 and 5.

The rest of the paper is structured as follows. Section 2 gives a description of the state-of-the-art works related to the topic of this paper. Section 3 describes the MAS proposed for guiding users in SNSs. Section 4 explains an experiment conducted with a SNS called Pesedia with known users at a laboratory. Section 5 describes the experiments performed with data from Twitter.com. Finally, Section 6 presents our conclusions and possible future lines of work.

2. Related Work

Since our goal is to build a MAS with agents that implement sentiment analysis, stress analysis, and combined analysis to guide users in on-line social environments in an attempt to prevent possible future issues by analyzing the user state, we will discuss previous approaches for sentiment and stress analysis as well as risk prevention and privacy aiding in SNSs. We will also review previous approaches on modeling the user state, where the state of the user is used by the system to make decisions like in our proposed system and works on MAS technology applied to SNSs to solve problems, to act as recommender systems, and to exploit the compatibilities of the MAS with SNSs. A quick discussion of works on applying MAS technologies to the Internet of Things (IoT) will be presented as well. To the best of our knowledge, there is no approach for guiding users through SNS that uses sentiment analysis, stress analysis and combined analysis on text messages when they are written to determine if the state of the user that writes them could generate a negative repercussion on the

SNS through this message and that warns the user when needed.

Sentiment analysis is a line of research that attempts to assess the recognition or detection of opinion, sentiments, evaluations, appraisal, attitude, and emotion in different kinds of media (e.g., written messages, images, audio, etc.) [12]. In the literature, there are four well-differentiated techniques for sentiment analysis in texts: document-level sentiment analysis, sentence-level sentiment analysis, aspect-based sentiment analysis, and comparative sentiment analysis [13]. The difference between them is the level of fine-grained analysis that we are going to implement, except for the case of comparative sentiment analysis, which is an exception: document-level analysis, which refers to sentiment detection associated to an entire document; sentence-level analysis, that is the analysis of the sentiment associated to sentences in texts; aspect-based analysis, or sentiment associated to specific aspects of the text, which can be sequences of words or other kinds of features, respectively; and comparative sentiment analysis refers to an exception to the other techniques. By using comparative sentences, we can learn which entities are the preferred ones, using comparative words for training the model by associating sentiment polarities to them [13]. We use aspect-based sentiment analysis in our approach, to be able to perform fine-grained sentiment analysis. This is explained more extensively in Section 3.

To assess aspect-based sentiment analysis, there are two main concerns or objectives to accomplish: the detection of aspects on which we are going to associate a sentiment polarity; and the sentiment classification itself, which is the process of labeling the aspects with a sentiment by using information from the data. There are hybrid approaches that carry out both objectives at once, and there are also several techniques for solving aspect-based sentiment analysis [10]. Aspect detection can be addressed by several different techniques: frequency-based methods use the terms with the most frequency in the training corpus to use them in the final aspect set for the model [14]; generative models are also used for detecting aspects such as Conditional Random Fields (CRF), which use a varied set of features [15]; non-supervised machine learning techniques are also used (e.g., Latent Dirichlet Allocation or LDA) [16]. In the case of sentiment classification, there are dictionary-based methods, which use a dictionary of aspects with assigned sentiment polarity and an algorithm for classifying texts with a polarity label based on the dictionary of aspects. For example, the most frequent polarity from the aspects found in the text under analysis, using the polarity associated with the aspects in the aspect set. The aspect set is trained, so polarity labels are assigned to its aspects using, for example, machine learning techniques; however, other techniques could be used [10]. There are machine learning approaches that use Support Vector Regression and other techniques to obtain the features for training the model, and we can also find non-supervised methods that use techniques such as relaxation labeling [10]. Finally, hybrid approaches detect aspects and assign sentiment polarities to them simultaneously [10]. Syntax-based methods obtain words associated with sentiment and extract other aspects by exploiting grammatical relations [17]. CRF are used to relate sentiments to aspects by means of extracting information from relations between words [18].

An algorithm called TensiStrength, which is derived from SentiStrength (sentiment strength detection) [6], uses a set of terms with stress labels and another set with relaxation labels that have been previously trained in the same way as dictionary-based methods of sentiment analysis. Those sets are then used to detect stress and relaxation levels in text sentences. The algorithm also implements exclamation mark detection within a sentence and boosts the stress or relaxation level depending on factors other than the pure analysis of aspects. To assign stress and relaxation labels to the aspect sets, an unsupervised learning method is used, which uses tweet messages annotated with strengths. The values are later refined with a hill-climbing method.

For the case of user state modeling, in [19], a nearest-neighbor collaborative approach for training user-specific classifiers was used, and the classifiers were combined later with user similarity measurements to solve a sentiment analysis task. A model for sentiment classification was used by Gao et al. in [20]. This model computes user and product-specific sentiment inclinations. A social-emotional model for detecting the social emotion in a group of entities was created by Rincon et al. in [21]. The authors modeled the emotions of the entities using the pleasure, arousal, and dominance (PAD) three-dimensional emotional space and an ANN to learn the emotion of a group of entities when an event just happened. In our case, the modeling of the state of the user is done by analyzing the sentiment, the stress levels and by performing a combined analysis on the messages being written in a SNS (which to the best of our knowledge is not performed in previous works). This allows us to capture more aspects of the psychological state of the user than using only one kind of analysis and helps us to determine whether or not this message could generate a negative repercussion or problem in the SNS.

With regard to aiding privacy in SNSs, in [22], improving privacy was addressed by designing the user interface specifically for that purpose, making the core features of privacy in the system visible to the users by inserting privacy reminders and customized privacy settings. In our case study, we use the text messages of users in a SNS to analyze them and extract the sentiment polarity and stress level in order to later be able to guide users in their experience and help their privacy by avoiding spreading information that may trigger privacy issues. An example of protecting users in a SNS by analyzing their sentiment is presented in [23]. The authors built an SNS that used adult image detection (pornography), a message classification algorithm, and sentiment analysis in the text messages to help the system ban users that were incurring in on-line grooming and cyber-bullying. To the best of our knowledge, even when there are systems in the literature that attempt to prevent problems in SNS by using sentiment analysis, none of them use sentiment analysis, stress analysis, and a combined analysis to analyze the messages and determine whether or not the user should be warned about posting a message.

There are several works with proposals applying MAS architectures for creating recommender systems for SNSs, or that simply use them for exploiting the compatibilities of the MAS model with the structure of SNSs (both have separated entities that interact between them and the system), and also there are works proposing the use of ANNs in such MAS architectures. Agent and multi-agent approaches are suggested in [24], which work as communication mediators between users and social groups in SNSs; a MAS architecture that uses a connectionist ontology which uses ANNs with input and output nodes associated with logic variables, and represents user behavior is presented in [25]. The ontology is constructed by monitoring user behavior, and later used for collaborative filtering recommendation, by computing inter-ontology similarities; in [26] a MAS architecture is used to compute reputation based on ratings of products and services in an e-tourism setting, using different agents in charge of different roles and an ANN for computing the reputation; the relations between MAS and SNSs and ways to use MAS technology to support SNSs that have been implemented, and others that could potentially be implemented in the future are discussed in [27]. Moreover, other works propose the use of ANNs in MAS architectures for solving problems. In [28], a MAS architecture uses trust and reputation of agents to give an indication of how much agents can be trusted as experts, employing certificated recommendations between agents, based on a level of assurance computed on the basis of signed transactions and witnessed transactions; in a task of production planning in [29], a MAS architecture employs an agent that exploits a rule base to determine the input that receives an ANN that outputs production orders. The agent using the rule base computes several characteristics of the task to be performed such as number of tools or resolution of the product, which are necessary for the ANN model to compute the final production order; an XML-based MAS architecture called MAST is presented in [30] for supporting business-to-customer e-commerce activities, by building, updating, and exploiting user personalized profiles by weighting the activities performed in B2C processes.

Works on MAS architectures applied to IoT ecosystems are also found in the literature. In [31], a series of works using agent-based technologies for implementing IoT ecosystems, and works in performing IoT simulations are presented, while also discussing the advantages and disadvantages of using agent-based technology for these purposes; an algorithm called CoTAG was designed in [32] for creating groups of agents based on information about reliability and reputation in the IoT environment. Credibility in SNSs is explored in several works as can be seen in [33], where it is shown that text analysis is employed effectively for this task. Nevertheless, semantic analysis of text and multimedia should be explored further, and studies on the area lack experiments with large datasets and high-performance algorithms, and there is also a lack of publicly available standard datasets. While all the works mentioned about the use of MAS technology on SNS and IoT ecosystems attempt to address important tasks to the better functioning of the on-line social environments and IoT environments, none of them address the task of detection of the user state for prevision of potential future issues in the system, helping users to prevent them, as we do in our proposal.

Several works perform sentiment polarity and stress level detection in the literature and in industry. There are also systems that try to model the user state and others that aim to improve the privacy of the users. Our proposed system aims not only to be able to detect the sentiment polarity state of the user, the stress level and a combined value at the moment he or she is interacting, but to also use this information in the best way possible to prevent future bad situations by warning or advising the user based on the mental-state model made from the analysis of his or her text data. For this reason, we created two different analyzers and an Advisor agent in the MAS that perform sentiment analysis, stress analysis and a combined analysis on text messages. We conducted experiments with data from Twitter.com to discover which analysis is able to predict a state of the user that best helps in predicting future negative outcomes in social environments.

3. System description

We designed the system as a MAS that helps users by analyzing the data from the written messages that they post on social media, using different agents to perform different kinds of analyses (sentiment, stress, or combined) to determine if there should be feedback such as a warning displayed to the users to protect them from potential negative outcomes that could arise from their interaction. We used the SPADE multi-agent platform [34] to implement the agents of the system proposed. This system can be integrated into different SNSs or other social platforms via web requests.

The MAS is built using agents that are in charge of the several roles that need to be performed. They communicate with each other using a messaging interface that is built into the SPADE platform, which is based on the FIPA-ACL [35] language. There are three types of agents. Two are presentation agents that are in charge of receiving the data from users and sending feedback from the MAS back to the users, respectively. There are also agents that perform analyses on data and generate advice and warnings (Sentiment and Stress analyzers and an Advisor agent). Finally, there is one persistence type of agent that controls the flux of data from the MAS to the database and vice versa. An overview of the architecture of the system is shown in Figure 1. The agents in charge of the analysis and feedback generation of the MAS are explained in more detail in the following sections.

3.1. Sentiment analyzer agent

[1] https://www.tensorflow.org
[2] https://keras.io

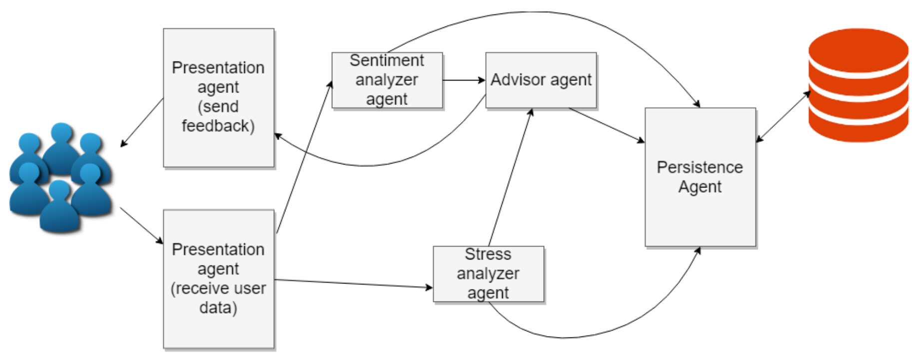

Figure 1: Architecture of the MAS

The Sentiment analyzer agent uses the text of short written messages as input and obtains a sentiment polarity from it as output, which is a qualitative value that is either negative or positive. In the case of the Sentiment analyzer agent, since we are only interested in knowing whether or not the message has a negative polarity label, the positive and neutral labels that can be found in the literature are grouped to represent one class and the negative label represents the second class. This agent calculates the sentiment polarity of text messages using the trained ANN when a user posts a message, and also sends the calculated polarity to the Advisor agent to potentially send feedback to the user as well as to the persistence agent to store the history of polarities. As stated above, the Sentiment analyzer agent is based on a feed-forward ANN model built using Tensorflow[1] and Keras[2] in Python, using embedding layers for modeling text sequences. We chose to use an ANN for our analyzers since it has been reported in [10] that supervised machine learning techniques have been shown to perform at state-of-the-art accuracies in the aspect-based sentiment analysis task. The architecture of the network, which is explained below, is shown in Figure 2.

First of all, a tokenizer that has been trained to convert words in Spanish to integers takes the input sentence and creates a vector of integers with it, using a mapping function from words to integers. The embedding layer then takes

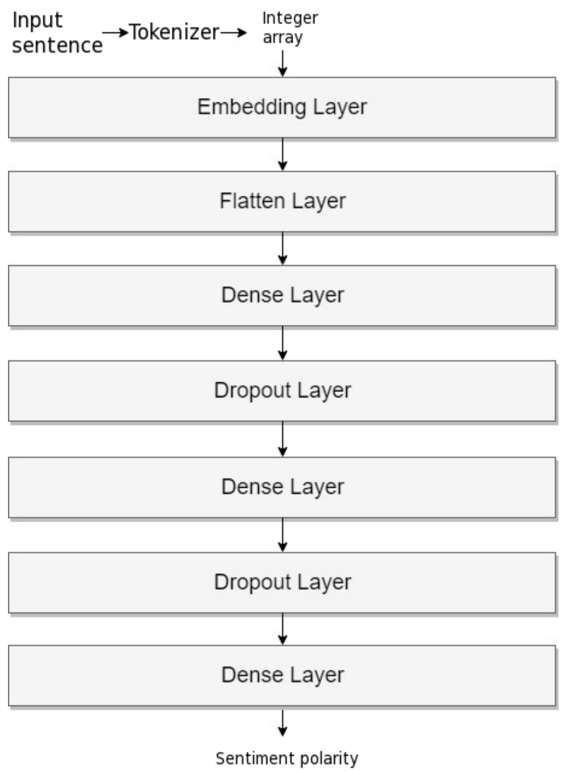

Figure 2: Architecture of the ANN for the Sentiment analyzer agent

this vector as input and gives the corresponding vectors with the embeddings associated with the texts as output. The embedding vectors are then given to the flatten layer, which converts this input into a flattened vector with one dimension that is fed to the first dense layer. A dropout layer follows the dense layer as a regularization mechanism with a dropout rate of 0.25, followed by another dense layer like the first one and a dropout layer with the 0.25 dropout rate again. The dropout rates were adjusted experimentally in order to obtain the best accuracy in the training of the ANN. The use of two pairs of dense and dropout layers instead of one was also found to give better results experimentally. Finally, a dense layer acts as the final layer of the network. All three dense layers use a sigmoid function as their activation function, and the two internal ones have a dimensionality of 64 in the output vector. The ANN uses binary cross-entropy as the loss function and an Adam optimizer [36]. Again, the activation functions, dimensionality, loss function, and optimizer are parameters that were adjusted experimentally in order to obtain the best accuracy in the training of the ANN.

The ANN has been trained and validated using a dataset of texts labeled with an emotion from a set of five possible emotions. This dataset was inspired in the PAD temperament model [37] (Happy, Bored, Relaxed, Anxious and Angry) and also labeled with a flag for low or high-stress levels. It was constructed by young people (both male and female) with ages ranging from 12 to 15 years old who used self-report. In other words, they were asked to label their messages with this information, but not forced to, so only the labeled messages are inserted into the dataset. To train and validate the model, a mapping from the five emotion labels in the dataset to two labels (positive or negative emotion) has been done. The mapping is as follows:

1. Happy: mapped as a positive sentiment.
2. Bored: mapped as a negative sentiment.
3. Relaxed: mapped as a positive sentiment.
4. Anxious: mapped as a negative sentiment.
5. Angry: mapped as a negative sentiment.

Mapping is applied to the five emotion original labels to only two labels (positive and negative sentiment) to the training dataset mentioned above. The network is then trained and validated using this binary-labeled data. Therefore its classification is binary to negative or positive sentiment. In the case of sentiment analysis performed in this work, aspect detection is performed during the training, when the tokenizer is used to fit the words in the training dataset by processing the messages and extracting words in order to latter assign integers to them to feed the embedding layer. The sentiment classification is done when training the ANN, so it assigns weights based on the labeled text messages of the dataset. In the validation process, an accuracy of 64.8 % was obtained.

When compared to the precision of 68.0 % to 77.2% found in state-of-the-art, supervised machine learning, aspect-based sentiment analysis [10], it is a little low but still close. To understand why the accuracy is a bit low, it is important to remember the following: we needed to map the labels from five states to two; we had a dataset for the training constructed by short texts from children ranging in age between twelve to fifteen years old; labels were made using self-report from the writer of each text, and the dataset was not very big (6,475 messages).

3.2. Stress analyzer agent

The Stress analyzer agent uses a similar ANN architecture and the same dataset as the Sentiment analyzer agent. However, the model is trained using the stress labels (low or high-stress level) found in the dataset for the training and validation. An image of the architecture of the ANN for this agent is shown in Figure 3.

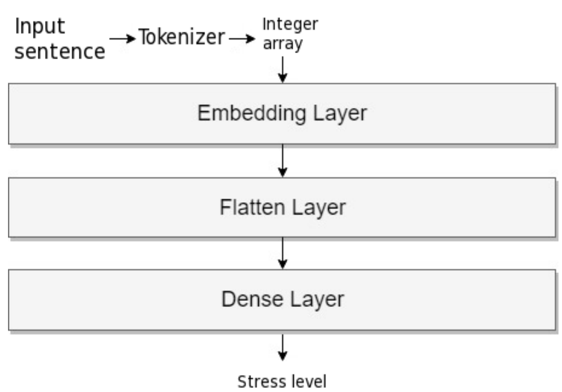

Figure 3: Architecture of the ANN for the Stress analyzer agent

This agent takes a text as input and classifies it with a low or high-stress level class label, using the model trained with the labeled dataset in the same way as the Sentiment analyzer works when users post text messages. It also sends the calculated stress-level label to the Advisor agent and the persistence agent. The difference in the architecture of the ANN of this agent from the one of the Sentiment analyzer agent is that it does not have the two pairs of dense layer and dropout layer that were found in the middle of the pipeline of the Sentiment analyzer. Experimentally, the performance was better for this model without one of them. Finally, an accuracy of 72.3 % was obtained in the validation process. When comparing this accuracy with the precision of 68.0 % to 77.2% found in state-of-the-art, supervised machine learning, aspect-based sentiment analysis [10], it can be observed that this classifier achieved state-of-the-art accuracies.

3.3. Advisor agent

The Advisor agent accomplishes two different tasks: it integrates the combined analysis and also generates warnings to give feedback to the users, if necessary. It obtains the information about sentiment polarity and stress level from the Sentiment analyzer and the Stress analyzer agents when a user posts a message and this message is sent to the MAS so that those values can be calculated. To compute this label, we assign a negative label to the message if we find either a high-stress label in the output of the Stress analyzer or a negative sentiment polarity label from the Sentiment analyzer. Otherwise, we assign a positive label. This process is shown in Figure 4. We used this version of combined analysis instead of the one that uses the intersection of messages detected by both analyzers (will be shown in the next section), because it was more inclusive (detects more negative messages, since the union is less restrictive), and we did not know which analyzer would perform best before performing the experiments. Finally, if the combined analysis assigns a negative label to the message, the Advisor agent generates a warning and sends it to the presentation agent in charge of communicating the feedback of the MAS to the user. This agent also stores the combined value calculated in the database via the persistence agent just like the Sentiment and Stress analyzers.

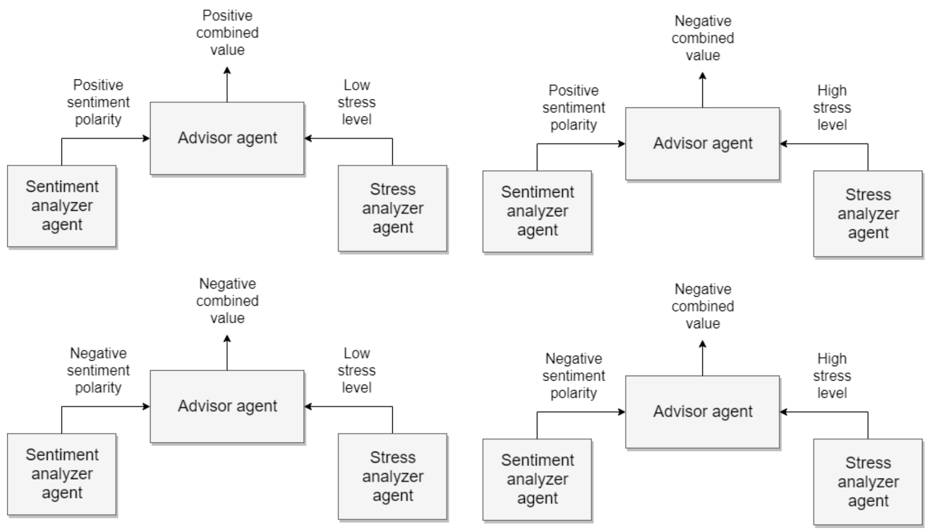

Figure 4: Combined analysis integrated in the Advisor agent labeling process

3.4. Example of the functionality of the analyzers and Advisor agents

Consider a scenario where a user in a SNS is about to publish a post on his or her wall. This message is sent to the presentation agent of the MAS, who receives it and sends it to the Sentiment and Stress analyzers to calculate the associated sentiment and stress labels. Then, the text message and labels computed are stored in the database and sent to the Advisor agent. The Advisor agent calculates the combined value and discovers that it is deemed negative (e.g., The high stress was not detected to be present, but a negative sentiment polarity was detected, thus giving a negative combined value). Then, this agent sends a notification of this negative status to the presentation agent, who sends it to the SNS, and the SNS displays a warning to the user to advise him or her to rethink his or her post. This way the user knows that the message that he or she is writing could lead to a possible bad outcome.

4. Experimentation with the social network Pesedia

We performed experiments with real users of a social network called Pesedia, with the proposed MAS integrated as functionality for analyzing the emotional state and stress levels of the users and advising them at the moment of posting messages on the network. This social network was used by children, who ranged in age between twelve and fifteen years old. Pesedia was made with the social networking engine Elgg[3]. The network is structured into diverse plug-ins that build functionalities from a base that is a generic social networking site. We conducted a set of experiments over two weeks. Our MAS was integrated into Pesedia and worked by recommending the erasure of messages if at the moment of posting they were deemed negative by the Advisor agent of the MAS. We created a test group and a control group to monitor the differences between using our MAS or not using it, which means that in the control group there was no advising functionality. The control group was composed of 76 children, and the test group was composed of 46 children. The total number of children participating was 122.

Our goal was to expose our MAS to a real-life environment with users that interact with it so that we could check its functionalities in real-time situations. For that purpose, we let children participate in Pesedia and interact with our system simultaneously, so they could provide feedback to the system. We also used a survey that the users of Pesedia filled out to understand how they felt about the feedback of the system and to know if they thought that the emotional state of the user affected their social interaction. Both the experiments conducted and the results of the surveys are presented below.

The following concepts are used in the metrics of these experiments:

- *positiveMsgs*: Number of messages from the group of people being analyzed that are detected as positive by the system (combined analysis).
- *negativeMsgs*: Number of messages from the group of people being analyzed that are detected as negative by the system (combined analysis).

[3]https://elgg.org/

- *totalMsgsGroup*: Total number of messages analyzed from a group in the experiment (either the control group or the test group).
- *totalMsgsWithReplies*: Total number of messages that generated replies in Pesedia analyzed.
- *msgConcSen*: Number of messages with the same emotional polarity as the one that is most present in their own replies.
- *msgConcStr*: Number of messages with the same stress level as the one that is most present in their own replies.
- *msgConcComb*: Number of messages with the same combined value as the one that is most present in their own replies.

The formulas for the metrics used in the experiments with Pesedia are the following:

- Percentage of positives (percentPositives): Percentage of messages with a detected positive state, generated in either the control group or the test group.

$$percentPositives = \frac{positiveMsgs}{totalMsgsGroup}$$

- Percentage of negatives (percentNegatives): Percentage of messages with a detected negative state, generated in either the control group or the test group.

$$percentNegatives = \frac{negativeMsgs}{totalMsgsGroup}$$

- Propagation of sentiment for known users (PSENKnown): Propagation of the sentiment polarity in the experiment with Pesedia users.

$$PSENKnown = \frac{msgConcSen}{totalMsgsWithReplies}$$

- Propagation of stress for known users (PSENKnown): Propagation of the stress levels in the experiment with Pesedia users.

$$PSTRKnown = \frac{msgConcStr}{totalMsgsWithReplies}$$

- Propagation of combined value for known users (PSENKnown): Propagation of the combined value in the experiment with Pesedia users.

$$PCOMBKnown = \frac{msgConcComb}{totalMsgsWithReplies}$$

The members of both the control and test groups started to use the social network over a period of two weeks that the experiments lasted. After this period ended, we launched laboratory experiments to analyze the results. These are listed below. The results of the experiments with Pesedia are reviewed following.

- For the first experiment, we took the messages from the database of the social network from both groups (the test and the control group) and analyzed them with the combined analysis to be able to compare the results.

- For the second experiment, since we asked the users to erase the messages considered negative, we searched the database for those messages to determine if the users had actually erased them.

- For the last experiment, we performed an analysis of all of the messages posted on the walls of the social network that were stored in the database from all of the groups of users, using one different analysis at a time. This way we could compare the propagation that the emotional polarity, the stress level, and the combined value had in the social network by comparing the results of the analysis on the messages with the predominant (most present) value obtained in the analysis of the replies that they generate.

In the first experiment, which is the comparison between the emotional state and stress levels of the test group and the control group, we calculated the combined analysis of our system in the text messages as percentNegatives and percentPositives for both groups and summarized the results in Table 1. As the table shows, there is a difference in the percentage of messages that the system detects as negative between one group and the other, with the control group being around 4.91 % higher in total percentage of negative messages detected than the test group, showing that there were fewer messages detected as negative in the group with the warnings.

In the second experiment, which is the comparison of whether or not the users really erased the messages that the system detected as negative, we discovered that, as a general trend, the users did not erase their messages despite receiving the alert message from the system. It must be taken into account that the goal of the system was to give feedback to users to guide them, so it does not perform persuasion techniques on users but instead warns them. Adding persuasion techniques to the feedback system might potentially lead to achieving users erasing the messages when the system warns them about them.

Finally, in the last experiment, we analyzed the propagation of three psychological states in a user when he or she interacts with the social network (sentiment polarity, stress level, and combined state) by comparing the state of the user who writes the post with the most frequent state of the users who reply to that post. In this case, we analyzed all of the data of Pesedia at the same time using one of the three analyses at a time. The results are summarized in Table 2. As the table shows, the Sentiment analyzer detects that there is

51.79 % propagation between the values of sentiment polarities of the original messages and the replies. The Stress analyzer indicates 52.81 % propagation, and the combined analysis shows 55.36 % propagation. The Sentiment and Stress analyzers and the combined analysis that has been explained in Section 3 obtained similar results of propagation, with the difference that the combined analysis performed about 2.55 % better in terms of propagation than the other analyzers. It should be taken into account that the data did not contain a large number of text messages (as it was generated in a short span of time by only 122 children), which may make the experiments less representative.

Table 1: Comparison of the analysis on the test and control groups

	percentagePositives	***percentageNegatives***
Group with alerts	38.07 %	61.93 %
Control group (No alerts)	33.16 %	66.84 %

Table 2: Comparison of the different analyses in all of the data generated during the experiments

PCsenKnown	***PCstrKnown***	***PCcombKnown***
51.7857 %	52.8061 %	55.3571 %

In addition to the performed experiments, we also gave a survey to the users of Pesedia in order to understand how they felt about the feedback that our MAS was giving them and also to know if they thought that the emotional state of the users affects their social interaction. The questions asked on the survey were:

1. Has the advice regarding the risk of publishing a message been useful to you?
2. Has the advice regarding the risk of publishing a message been annoying to you?
3. Have you taken into account the advice regarding the risk of publishing a message?

4. You did not receive any advice or alerts, would you have preferred that the social network informed you that your publications might be potentially risky?
5. Do you believe that privacy problems can arise from publishing a post?
6. Do you believe that your emotional state influenced the repercussions of your messages?

And the possible answers were:

- Yes
- No
- Does not apply to me

There were two exceptions; the last two questions only had the first two options (yes/no) because they are general opinion questions.

Table 3: Summary of the results of the surveys

Question	***Yes***	***No***	***Does not apply***
1	9.43 %	9.43 %	81.13 %
2	7.55 %	13.21 %	79.25 %
3	10 %	10 %	80 %
4	69.81 %	26.42 %	3.77 %
5	65.56 %	34.44 %	0 %
6	38.89 %	61.11 %	0 %

The summarized results of the surveys are presented in Table 3. It can be observed that even when not many users seem to be getting alerts, they wish that the social network alerted them about potential risks and they thought that problems could arise from publishing a post. In the future, we aim to create better feedback for the user. Despite the general trend of users to think that a problem can arise from publishing a post, in general, they answered that they don't think that their emotional state has influenced their messages. The few users that received alerts were equally divided in opinions about whether or not the alerts were useful to them or if they had taken them into account. These results are in line with the second experiment since in general people were not erasing the messages when the warnings were shown. Finally, the number of users that answered that they received alerts and that the alerts were not annoying to them was close to double the ones who thought they were annoying (13.21 % vs 7.55 %).

5. Experimentation with data from Twitter

Since the data we collected in the experiments with the Pesedia SNS was not large (as it was generated in a short span of time by only 122 children), and one of our intentions was to discover how the system worked if it was used in different environments, we conducted experiments with data from Twitter.com. The goal of these experiments is to be able to decide what analysis or analyses should be considered to be more informative than others and in which cases a warning should be raised in the Advisor agent of our system. To achieving this, we compare the values obtained using the sentiment, stress, and combined analyses on the text messages with the values obtained for their replies. This is what we call checking the propagation to the replies of the state that is obtained with the different analyses. We aim to discover if it happens, to what extent, and in which cases. This is important since we would be able to create more useful feedback for the users navigating in SNSs if we could detect negative user states that could potentially propagate more in the network. The analyzers used in this experimentation are the same as the ones used in the experiments with Pesedia users (with the exception of the different modality of combined analysis, which will be presented later in this section), which are the ones shown in Section 3. We designed our system as a guiding system for on-line social environments where young people interact, therefore we used a dataset with data from teenagers for training the models. The experiments with data from Twitter.com aim to discover which is the best way to use the analyzers for preventing negative situations, and we used our models to be able to build a better guiding system in the future. To conduct the experiments, we extracted data from Twitter.com to create three corpora of tweets.

The three corpora (short text messages from the SNS Twitter.com) have been extracted using the Twitter API for streaming tweets. These corpora have no geographic limitation (they can be composed of messages from people at different locations around the globe), and each one has a different theme (e.g., politics, leisure). The messages in the corpora are composed of tweets that are replies to other tweets since we need replies in order to study the relationship between the detected emotional state and the stress level of the tweets with their replies. Moreover, the messages are in Spanish since the tokenizers that convert text to integers to feed the ANNs that perform sentiment, stress, and combined analyses currently only recognize words in Spanish. The corpora are the following:

- Podemos: This is a corpus of messages that are related to the political party Podemos. It is a very large corpus with 223,458 tweets.
- Star Wars: This is a corpus that is related to the Star Wars franchise, and is, therefore, a leisure corpus. It contains only 22,543 tweets.
- El Confidencial: This is a corpus composed of tweets about the digital newspaper *El Confidencial*, located in Spain, which is specialized in economic, financial, and political news. It contains 482,633 tweets.

We carried out the experimentation with the three analyses two times, using one different variation of the combined analysis at a time. The difference between the two is that one performs a combination of the information of the Sentiment analyzer and the Stress analyzer using the union of the sets of messages detected with a high level of stress and the messages detected with negative sentiment polarity for assigning a negative label to messages, and the other uses the intersection of those two sets. The analysis that uses the union of sets is called the 'or' version of the combined analysis, and the one using the intersection is called the 'and' version. The 'or' version is the one currently used in the MAS, since as mentioned in Section 3.3 it was the first implementation made before conducting the experiments with Twitter.com and testing with another version, and it is more inclusive in the sense that it is less restrictive detecting messages. This may be changed in a future version of the system.

In the experiments, we proceeded in the following way: We process the tweets assigned to the experiment one by one. First of all, we check if the tweet that generated the reply being processed has been analyzed previously, if so then only the sentiment analysis and stress analysis on the reply are computed. Otherwise, we use the twitter API to search for the message that generated the reply. Then we calculate the sentiment polarity and stress level of both messages and store them together. When all of the tweets assigned to the experiment are processed, for all of the tweets that generated replies, we do the following:

1. Compute its combined value using the sentiment polarity and stress level and the combined analysis (using either the 'or' version or the 'and' version of the analysis).
2. Compute the predominant sentiment polarity in the replies of the tweet (predominant as the most present sentiment polarity).
3. Compute the predominant stress level in the replies of the tweet.
4. Compute the predominant combined value of the replies using the previously obtained predominant sentiment polarity and predominant stress level in the same way that the combined analysis works with the sentiment polarity and stress level of a single tweet.

When we have finished with the above process, we proceed to compare the individual values of sentiment, stress, and the combined value of the original tweets with the predominant values in the replies. This way, we know if the sentiment, stress level, or combined value has propagated from a tweet to its replies. Finally, we calculate the percentage of the tweets that generated replies with a predominant or most present sentiment value that was the same calculated for them and store it as a final result. We do the same for the stress level and for the combined value, obtaining three results from the experiment. We explain the calculation in more detail showing the metrics used following.

First, we present the concepts that are used in the calculus of the metrics:

- *total_tweets*: Total number of Tweets that generated analyzed replies.
- *propagated_tweets_sen*: Tweet messages with the same emotional polarity as the predominant emotional polarity calculated in their replies.
- *propagated_tweets_str*: Tweet messages with the same stress level as the predominant stress level calculated in their replies.
- *propagated_tweets_comb*: Tweet messages with the same combined value (as the output of the combined analysis) as the predominant combined value calculated in their replies.

The formulas for the calculation of the metrics that we use in the experiments are the following:

- Propagation of the sentiment polarity (PSEN): proportion of propagated_tweets_sen in total_tweets.

$$PSEN = \frac{propagated\_tweets\_sen}{total\_tweets}$$

- Propagation of the stress level (PSTR): proportion of propagated_tweets_str in total_tweets.

$$PSTR = \frac{propagated\_tweets\_str}{total\_tweets}$$

- Propagation of the combined value (PCOMB): proportion of propagated_tweets_comb in total_tweets.

$$PCOMB = \frac{propagated\_tweets\_comb}{total\_tweets}$$

In order to analyze if there were differences in the propagation of the state detected by the different analyses caused by the number of tweets used in an experiment, we designed the experiments with the different corpora as a set of groups of experiments for each corpus, using a different number of tweets in each group, which we call partition size. Therefore, we performed experiments with a different number of tweets, using different parts of the corpus. In the case of the Podemos corpus, since it is a large corpus we decided to make six different partition sizes (1/4, 1/8, 1/16, 1/32, 1/64, 1/128 of the tweets of the corpus, respectively). For avoiding using the same data in different experiments, we performed groups of 4 experiments for each partition size, since the biggest partition size only allows a maximum number of four, and we decided to perform the same amount of experiments for each partition size. The set of six groups of experiments was also performed two times, one time using the 'or' version of the combined analysis and one time using the 'and' version. The final results of the experiments are shown in Table 4 for the experiments with the 'or' version of the combined analysis and in Table 7 for the 'and' version.

For the case of the Star Wars corpus, we used only four different partition sizes and performed three experiments for each partition size. We did it this way because the number of tweet messages was not high (22,543 tweets). We proceeded in the same way as we did with the case of the Podemos corpus when designing the experiments, with the only difference that in this case, the biggest partition size was 1/3. The final results of the experiments with this corpus are shown in Table 5 for the experiments with the 'or' version of the combined analysis and in Table 8 for the 'and' version. Finally, for the case of El Confidencial corpus, since it is a large corpus (482,633 total tweet messages), we used six partition sizes and four experiments for each of them as in the Podemos corpus. The results for these experiments are shown in Table 6 for the experiments with the 'or' version of the combined analysis and in Table 9 for the 'and' version.

The results for the experiments with the Podemos, Star Wars, and El Confidencial corpora when the 'or' version of the combined analysis was used are shown in Figures 5, 6, and 7, respectively. For the experiments using the 'and' version of the combined analysis, the results are shown in Figures 8, 9, and 10. The values for all of the experiments performed for each partition size in each corpus have been averaged to be represented as one single dot in the figures (e.g., the four experiments performed with 1/4 partition size for the Podemos corpus are represented as one dot with the average of the four values). The figures show the values for each analysis represented separately, and the legends represent the following metrics:

- SA and Stress A: PCOMB.
- SA: PSEN.
- Stress A: PSTR.

To assess whether or not the observed differences of propagation in the state detected by the different analyses are statistically significant, a t-test was executed for each pair of analyses, and for each experiment. The alpha type one error that is chosen for rejecting the null hypothesis of no difference in the means is 0.05. The results are shown in Table 10, where the t-value, the critical t-value for the two-tailed test, and the P-value or $P(T \leq t)$ for the two-tailed test are shown for each t-test performed. The results of the several t-tests and the general results for the experiments are analyzed following.

Table 4: Experimentation with the Podemos corpus using the Sentiment and Stress analyzers and the 'or' combined analysis

Partition size	***Experiment***	***PSEN***	***PSTR***	***PCOMB***
1/128 of replies	1	0.5117	0.5418	0.5284
	2	0.5305	0.5663	0.5233
	3	0.5423	0.6038	0.55
	4	0.5187	0.5841	0.5514
1/64 of replies	1	0.5225	0.5915	0.5385
	2	0.5091	0.5212	0.4545
	3	0.5093	0.5186	0.4596
	4	0.5449	0.5415	0.515
1/32 of replies	1	0.5296	0.5551	0.4987
	2	0.5406	0.5058	0.4984
	3	0.5393	0.5393	0.5223
	4	0.512	0.5	0.5051
1/16 of replies	1	0.5208	0.5378	0.512
	2	0.5229	0.5326	0.5296
	3	0.5308	0.5374	0.5215
	4	0.5446	0.5401	0.5166
1/8 of replies	1	0.5302	0.5509	0.5296
	2	0.5456	0.5334	0.5141
	3	0.5343	0.5399	0.5173
	4	0.5402	0.5432	0.531
1/4 of replies	1	0.5339	0.5426	0.5215
	2	0.5423	0.539	0.527
	3	0.5132	0.5276	0.5078
	4	0.5225	0.54	0.5071

For the Podemos corpus, it can be observed that there are small differences for the propagation detected by the different analyses except for the case of the 'and' version of the combined analysis. The Stress analyzer performed significantly better than the Sentiment analyzer in terms of propagation in the experiments with the 'or' combined analysis. The t-value was -2.3718 and the critical t-value was 2.0369, so the difference is significant for the chosen alpha

Table 5: Experimentation with the Star Wars corpus using the Sentiment and Stress analyzers and the 'or' combined analysis

Partition size	***Experiment***	***PSEN***	***PSTR***	***PCOMB***
1/24 of replies	1	0.5617	0.5185	0.5556
	2	0.4909	0.497	0.4848
	3	0.5535	0.5283	0.5283
1/12 of replies	1	0.516	0.5209	0.5209
	2	0.5638	0.5904	0.5319
	3	0.5229	0.5443	0.4924
1/6 of replies	1	0.4742	0.5274	0.5032
	2	0.5258	0.5619	0.5052
	3	0.5639	0.54	0.5349
1/3 of replies	1	0.5129	0.5224	0.5017
	2	0.5571	0.5483	0.5237
	3	0.5234	0.5781	0.5144

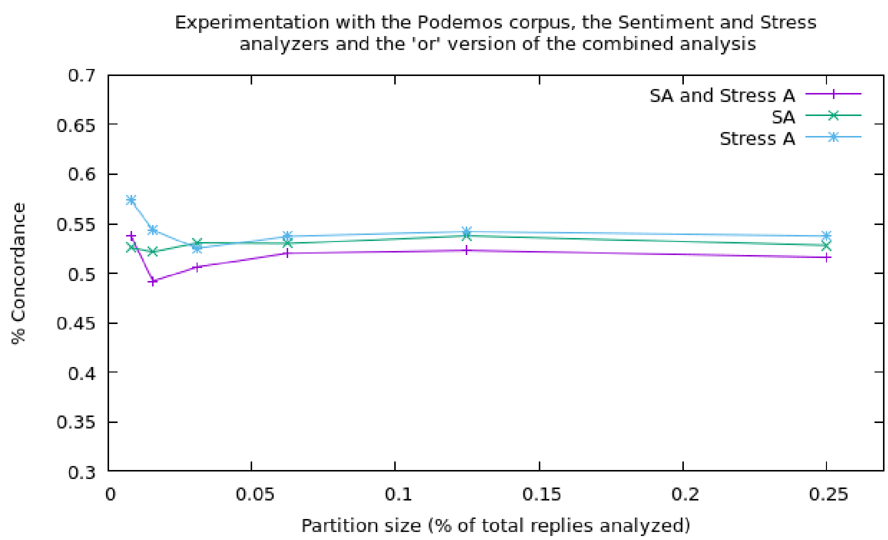

Figure 5: Results of the experiments with the Podemos corpus for the Sentiment and Stress analyzers and the 'or' combined analysis

0.05, with this difference being about 1.5%. The same happened in the experiments with the 'and' combined analysis. Again, there was a significant difference

Table 6: Experimentation with the El Confidencial corpus using the Sentiment and Stress analyzers and the 'or' combined analysis

Partition size	***Experiment***	***PSEN***	***PSTR***	***PCOMB***
1/128 of replies	1	0.5466	0.5552	0.4897
	2	0.5652	0.5786	0.505
	3	0.5411	0.5507	0.4952
	4	0.5314	0.569	0.4623
1/64 of replies	1	0.5557	0.5547	0.496
	2	0.538	0.5642	0.508
	3	0.5559	0.5798	0.5124
	4	0.5311	0.5697	0.5019
1/32 of replies	1	0.527	0.558	0.4791
	2	0.5581	0.5682	0.5168
	3	0.5407	0.5518	0.4955
	4	0.56	0.5493	0.5202
1/16 of replies	1	0.5351	0.5704	0.4936
	2	0.5373	0.559	0.4958
	3	0.5548	0.5676	0.5156
	4	0.5449	0.5668	0.5109
1/8 of replies	1	0.5407	0.5507	0.5019
	2	0.5545	0.5817	0.5218
	3	0.5695	0.5746	0.5236
	4	0.5619	0.5862	0.5253
1/4 of replies	1	0.5608	0.5657	0.5134
	2	0.5575	0.574	0.5177
	3	0.5527	0.5716	0.5111
	4	0.5526	0.5666	0.5081

of about 1.5%. The combined analysis in the 'or' version performed worse than the former ones, with this difference being significant according to the t-tests performed. It was about 3% worse than the Stress analyzer, which performed better than the Sentiment analyzer in this corpus. Since the Stress analyzer has higher accuracy detecting stress levels than the Sentiment analyzer detecting sentiment polarities (approximately 7.5%), it is not surprising that the Stress analyzer is able to detect a state that propagates more to the replies, even when

Table 7: Experimentation with the Podemos corpus using the Sentiment and Stress analyzers and the 'and' combined analysis

Partition size	***Experiment***	***PSEN***	***PSTR***	***PCOMB***
1/128 of replies	1	0.4797	0.5169	0.5709
	2	0.5201	0.5174	0.6139
	3	0.5151	0.5783	0.6175
	4	0.468	0.5181	0.5766
1/64 of replies	1	0.4974	0.5305	0.5986
	2	0.5092	0.5767	0.6258
	3	0.5173	0.5442	0.6154
	4	0.5704	0.5798	0.6784
1/32 of replies	1	0.5158	0.5397	0.6137
	2	0.5087	0.5303	0.6104
	3	0.532	0.5343	0.6042
	4	0.5377	0.5073	0.5947
1/16 of replies	1	0.5328	0.5538	0.6238
	2	0.5284	0.5365	0.6026
	3	0.5293	0.535	0.6073
	4	0.5384	0.5348	0.6165
1/8 of replies	1	0.523	0.5323	0.6143
	2	0.5305	0.5359	0.621
	3	0.5319	0.5421	0.6202
	4	0.5394	0.5355	0.6049
1/4 of replies	1	0.523	0.5387	0.6107
	2	0.5366	0.5266	0.6059
	3	0.5206	0.5358	0.6073
	4	0.5275	0.5353	0.6158

the difference is small. The combined analysis in the 'or' version detects a state that may be harder to track (the state detected in the replies depends on the state detected by both the Sentiment and Stress analyzers, and it can be a negative state if the analyzers detect either negative polarity or a high-stress level), thus potentially leading to less propagation. When using the 'and' version of the combined analysis, it can be observed that there is a difference of around 7.4% of propagation in favor of this analysis compared to the best of both the

Table 8: Experimentation with the Star Wars corpus using the Sentiment and Stress analyzers and the 'and' combined analysis

Partition size	***Experiment***	***PSEN***	***PSTR***	***PCOMB***
1/24 of replies	1	0.5758	0.7273	0.7576
	2	0.6471	0.5098	0.6078
	3	0.5	0.5227	0.6136
1/12 of replies	1	0.5707	0.544	0.6
	2	0.599	0.5681	0.6375
	3	0.5526	0.6029	0.6555
1/6 of replies	1	0.5516	0.5497	0.6341
	2	0.559	0.5864	0.664
	3	0.5449	0.5612	0.6378
1/3 of replies	1	0.5374	0.5522	0.6283
	2	0.5532	0.5728	0.6334
	3	0.5331	0.5912	0.65

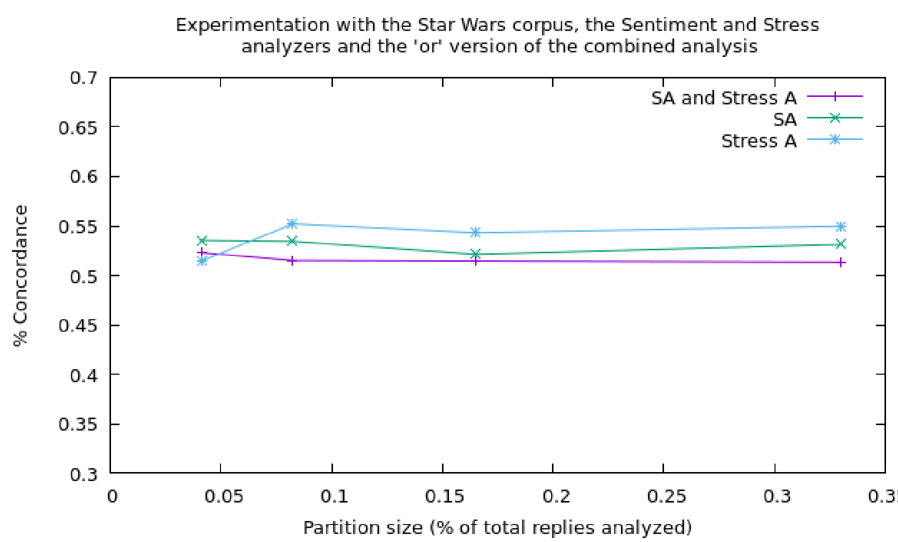

Figure 6: Results of the experiments with the Star Wars corpus for the Sentiment and Stress analyzers and the 'or' combined analysis

Sentiment and Stress analyzers. The difference between the state detected by the different analyses was again significant according to the respective t-tests.

Table 9: Experimentation with the El Confidencial corpus using the Sentiment and Stress analyzers and the 'and' combined analysis

Partition size	***Experiment***	***PSEN***	***PSTR***	***PCOMB***
1/128 of replies	1	0.534	0.562	0.6632
	2	0.5544	0.5882	0.6898
	3	0.5547	0.5566	0.6472
	4	0.5266	0.557	0.6578
1/64 of replies	1	0.5485	0.5711	0.6573
	2	0.5353	0.5521	0.6495
	3	0.5374	0.5816	0.679
	4	0.5511	0.5532	0.6375
1/32 of replies	1	0.5313	0.5686	0.6641
	2	0.5502	0.5742	0.6593
	3	0.533	0.5493	0.6521
	4	0.5418	0.5446	0.6686
1/16 of replies	1	0.5365	0.5643	0.6511
	2	0.5506	0.5572	0.6589
	3	0.5328	0.5771	0.6589
	4	0.5618	0.5591	0.6565
1/8 of replies	1	0.5599	0.5651	0.6665
	2	0.5538	0.5703	0.6661
	3	0.5676	0.5701	0.6702
	4	0.5641	0.5693	0.6641
1/4 of replies	1	0.5486	0.5641	0.6588
	2	0.5569	0.5715	0.6616
	3	0.5557	0.5652	0.6626
	4	0.56	0.5622	0.6651

Using the information of being detected as having both negative polarity and a high-stress level propagates better to the replies. This may be because the users that reply are influenced by both states of the user who posts the original message (high stress and negative sentiment polarity). Also, being detected as having a negative state by two different analyzers may mitigate the probability of being a false positive.

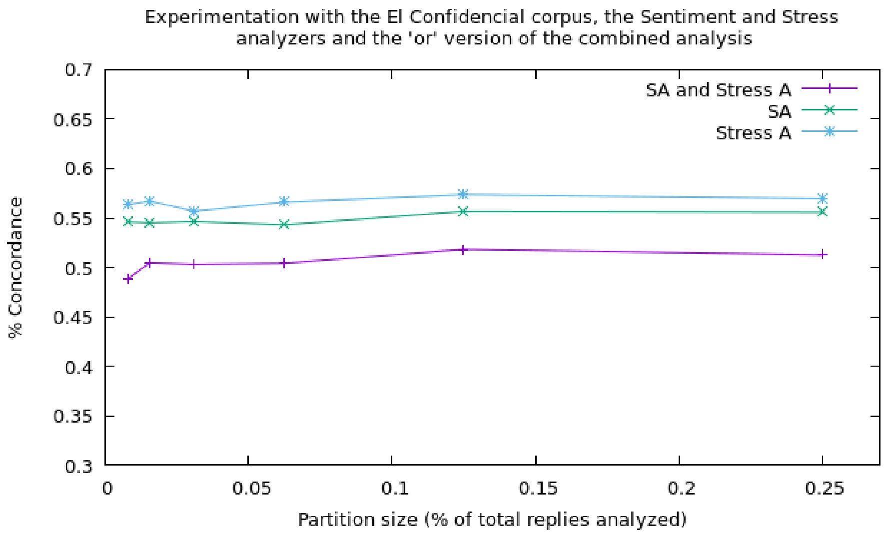

Figure 7: Results of the experiments with the El Confidencial corpus for the Sentiment and Stress analyzers and the 'or' combined analysis

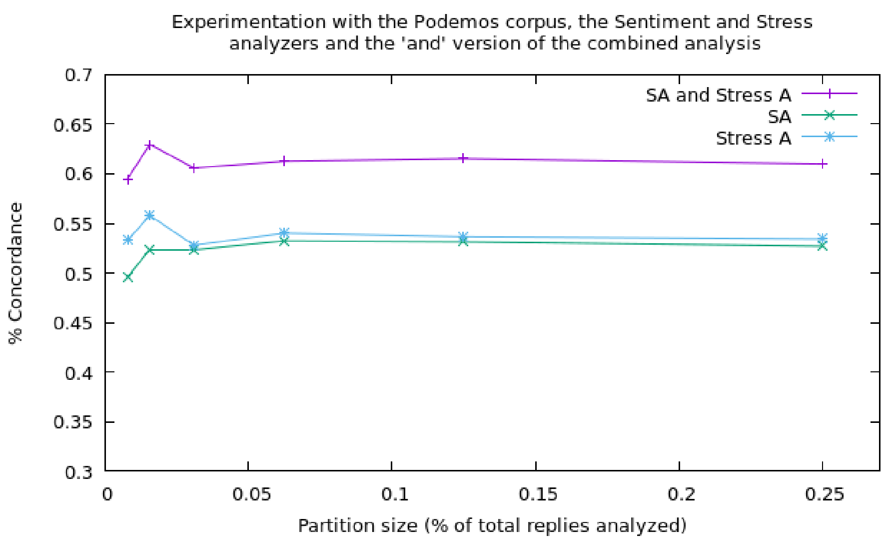

Figure 8: Results of the experiments with the Podemos corpus for the Sentiment and Stress analyzers and the 'and' combined analysis

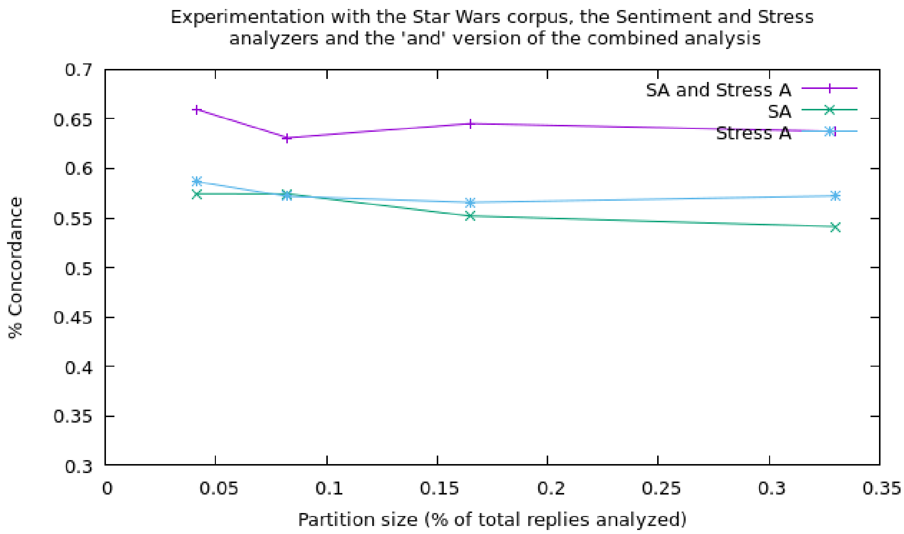

Figure 9: Results of the experiments with the Star Wars corpus for the Sentiment and Stress analyzers and the 'and' combined analysis

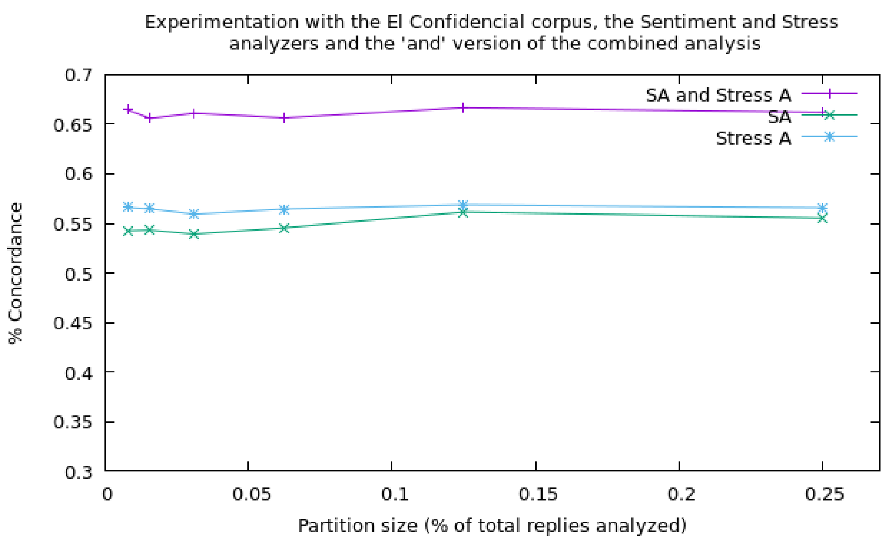

Figure 10: Results of the experiments with the El Confidencial corpus for the Sentiment and Stress analyzers and the 'and' combined analysis

Table 10: Results of t-test launched for comparing the significance of the difference observed in the propagation of the state detected by the analyzers

Experiments	*Results of propagation (compared)*	*t value*	*t critical value*	$P(T \leq t)$
Experiments with the Podemos corpus, the Sentiment and Stress analyzers and the 'or' combined analysis	PSEN and PSTR	-2.3718	2.0369	0.0239
	PSEN and PCOMB	2.6389	2.0345	0.0126
	PSTR and PCOMB	3.959	2.0154	0.0003
Experiments with the Star Wars corpus, the Sentiment and Stress analyzers and the 'or' combined analysis	PSEN and PSTR	-1.1615	2.086	0.2591
	PSEN and PCOMB	1.4483	2.1199	0.1668
	PSTR and PCOMB	3.0133	2.1098	0.0078
Experiments with the El Confidencial corpus, the Sentiment and Stress analyzers and the 'or' combined analysis	PSEN and PSTR	-5.2401	2.0167	4.5954E-06
	PSEN and PCOMB	10.8067	2.0181	1.0525E-13
	PSTR and PCOMB	16.0047	2.0227	9.8335E-19
Experiments with the Podemos corpus, the Sentiment and Stress analyzers and the 'and' combined analysis	PSEN and PSTR	-2.7599	2.0154	0.0084
	PSEN and PCOMB	-16.4879	2.0154	1.8888E-20
	PSTR and PCOMB	-13.9488	2.0154	9.2973E-18
Experiments with the Star Wars corpus, the Sentiment and Stress analyzers and the 'and' combined analysis	PSEN and PSTR	-0.0788	2.093	0.938
	PSEN and PCOMB	-5.7223	2.1314	4.0383E-5
	PSTR and PCOMB	-6.972	2.1009	1.639E-06
Experiments with the El Confidencial corpus, the Sentiment and Stress analyzers and the 'and' combined analysis	PSEN and PSTR	-5.0261	2.0167	9.2932E-6
	PSEN and PCOMB	-34.0411	2.0154	3.0008E-33
	PSTR and PCOMB	-30.8196	2.0154	2.0049E-31

The results of the t-tests for the Star Wars corpus show that only the difference between the PSTR and PCOMB metrics was significant for the experiments with the 'or' version of the combined analysis. The results of the t-tests also showed a significant difference between the PSEN and PCOMB metrics and between the PSTR and PCOMB metrics for the experiments with the 'and' combined analysis. A difference was found between the PSEN and PSTR metrics in favor of PSTR, but this difference was not statistically significant. The combined analysis performed worse in the 'or' version (about 3% compared to the Stress analyzer, which was the best of both the sentiment and Stress analyzers). However, about 7.3% better in the 'and' version compared to the Stress analyzer. Finally, the behavior of the analyses on the El Confidencial corpus resulted similarly to that of the Podemos corpus. Again, the differences between each pair of propagations were shown to be significant according to the t-tests. A small difference between the Sentiment analyzer and the Stress analyzer was found in favor of the Stress analyzer (about 2% in both the experiments with the 'or' and the 'and' combined analysis). The combined analysis (the PCOMB metric) in the 'and' version performed better (about 9.5% better than the PSTR metric, the best of both the PSTR and PSEN metrics), and in the 'or' version performed worse (about 6% worse than the PSTR metric).

In general, it can be observed that the Sentiment and Stress analyzers can separately and successfully predict a state of the user that propagates to the replies, which is shown by the metrics PSEN and PSTR. This is a general trend in the three corpora and in all of the experiments. The combined analysis has also shown this trend in both versions ('or' and 'and'). Nevertheless, there are differences between the two versions, as previously discussed in this section. In terms of propagation, we obtained a small difference in favor of the Stress analyzer over the Sentiment analyzer which may be due to the better accuracy of the Stress analyzer in detecting high levels of stress compared to the accuracy of the Sentiment analyzer detecting negative sentiment polarity. The 'or' version of the combined analysis performed slightly worse than the Sentiment and Stress analyzers, but the 'and' version performed better: about 7.4% of concordance more than the best of the Sentiment and Stress analyzers in the Podemos corpus; about 7.3% in the Star Wars corpus; and about 9.5% in the El Confidencial corpus. This means that we should expect to detect a user state that would propagate more to the replies if the 'and' version of the combined analysis detects the message as negative (high stress and negative sentiment polarity). This version of combined analysis could work as an additional source of information that is integrated into the Advisor agent that helps in deciding whether or not to advise the user. This is because it may lead to detecting user states in the messages that would have a greater probability of propagating in the network.

6. Conclusions and future work

In this work, a MAS for protecting and guiding users through the analysis of their emotional state and stress levels has been presented. The MAS integrates analyzers that use text data from users to determine their sentiment polarity

(Sentiment analyzer), stress level (Stress analyzer), and a combined analysis that uses both outputs, proposing two different forms of it (the 'or' and the 'and' version of the combined analysis). The analyzers are created using ANNs and the Tensorflow[1] and Keras[2] libraries for machine learning. The MAS also incorporates an Advisor agent that performs the combined analysis, generates warnings, and sends them as feedback to the users if necessary. This system works together with a social platform such as a SNS and guides users through their experience to protect them from future issues that could arise from the interaction. It takes the text messages in the social platform and analyzes them with the three different analyses to give advice (or not) if the message is deemed negative. We performed two different types of experiments: an experiment with a real SNS using our MAS to test it in a real-life environment, and an experiment with data from Twitter.com to determine which analysis would be more informative for the Advisor agent.

With regard to the experiments with the Pesedia social network, the control group generated more messages that were determined to be negative by the analysis than the test group that received the feedback, which is in line with the goal of the system. Also, in general, the users did not erase their messages despite receiving the alert message from the system recommending it. The addition of persuasion techniques could potentially help in getting users to erase the messages. Moreover, we detected that the 'or' combined analysis predicted a state of the user that propagated more to the replies than the state detected by the Sentiment analyzer and the state detected by the Stress analyzer, but with a small difference. We also gave a survey to the users of Pesedia in order to understand how they felt about the feedback of the system and if they thought that their emotional state had affected the repercussions of their messages.

[1]https://www.tensorflow.org

[2]https://keras.io

From the experiments with Twitter.com data, we discovered that the three analyses are able to detect a state of the user that posts a message that propagates to its replies. We observed a small difference in favor of the Stress analyzer over the Sentiment analyzer in terms of propagation. This may be due to the better accuracy of the Stress analyzer compared to the Sentiment analyzer. We also observed that the 'and' version of the combined analysis performed better than any other analysis in terms of propagation, with a greater difference than the case of the Stress analyzer compared to the Sentiment analyzer. Finally, the 'or' version of the combined analysis performed worse than the sentiment analysis, the stress analysis and the 'and' version of the combined analysis in terms of propagation.

For future lines of work, we plan to develop new agents that are capable of new types of analyses using other sources of information (e.g., typing patterns), and we will perform experiments to discover what analyses work best at detecting user states that propagate more to the other users in the network. We also aim to create better feedback for the users in the system by using different widgets and alerts, or adding persuasion techniques to the feedback.

Acknowledgments. This work was supported by the project TIN2017-89156-R of the Spanish government.

References

[1] E. Vanderhoven, T. Schellens, R. Vanderlinde, M. Valcke, Developing educational materials about risks on social network sites: a design based research approach, Educational technology research and development 64 (3) (2016) 459–480.

[2] S. De Moor, M. Dock, S. Gallez, S. Lenaerts, C. Scholler, C. Vleugels, Teens and ict: Risks and opportunities. belgium: Tiro (2008).

[3] S. Livingstone, L. Haddon, A. Görzig, K. Ólafsson, Risks and safety on the internet: the perspective of european children: full findings and policy implications from the eu kids online survey of 9-16 year olds and their parents in 25 countries (2011).

[4] E. Vandenhoven, T. Schellens, M. Valacke, Educating teens about the risks on social network sites, Media Educational Research Journal 43 (22) (2014) 123–131.

[5] J. M. George, E. Dane, Affect, emotion, and decision making, Organizational Behavior and Human Decision Processes 136 (2016) 47–55.

[6] M. Thelwall, Tensistrength: Stress and relaxation magnitude detection for social media texts, Information Processing & Management 53 (1) (2017) 106–121.

[7] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, A. Kappas, Sentiment strength detection in short informal text, Journal of the American Society for Information Science and Technology 61 (12) (2010) 2544–2558.

[8] E. Christofides, A. Muise, S. Desmarais, Risky disclosures on facebook: The effect of having a bad experience on online behavior, Journal of adolescent research 27 (6) (2012) 714–731.

[9] G. Aguado, V. Julian, A. Garcia-Fornes, Towards aiding decision-making in social networks by using sentiment and stress combined analysis, Information 9 (5) (2018) 107.

[10] K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Transactions on Knowledge and Data Engineering 28 (3) (2016) 813–830.

[11] J. A. Bordera, Pesedia. red social para concienciar en privacidad (2016).

[12] B. Liu, Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, 2012.

[13] R. Feldman, Techniques and applications for sentiment analysis, Communications of the ACM 56 (4) (2013) 82–89.

[14] M. Hu, B. Liu, Mining opinion features in customer reviews, in: AAAI, Vol. 4, 2004, pp. 755–760.

[15] N. Jakob, I. Gurevych, Extracting opinion targets in a single-and cross-domain setting with conditional random fields, in: Proceedings of the 2010 conference on empirical methods in natural language processing, Association for Computational Linguistics, 2010, pp. 1035–1045.

[16] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of machine Learning research 3 (Jan) (2003) 993–1022.

[17] T. Nasukawa, J. Yi, Sentiment analysis: Capturing favorability using natural language processing, in: Proceedings of the 2nd international conference on Knowledge capture, ACM, 2003, pp. 70–77.

[18] F. Li, C. Han, M. Huang, X. Zhu, Y.-J. Xia, S. Zhang, H. Yu, Structure-aware review mining and summarization, in: Proceedings of the 23rd international conference on computational linguistics, Association for Computational Linguistics, 2010, pp. 653–661.

[19] Y. Seroussi, I. Zukerman, F. Bohnert, Collaborative inference of sentiments from texts, in: International Conference on User Modeling, Adaptation, and Personalization, Springer, 2010, pp. 195–206.

[20] W. Gao, N. Yoshinaga, N. Kaji, M. Kitsuregawa, Modeling user leniency and product popularity for sentiment classification, in: Proceedings of the Sixth International Joint Conference on Natural Language Processing, 2013, pp. 1107–1111.

[21] J. Rincon, F. de la Prieta, D. Zanardini, V. Julian, C. Carrascosa, Influencing over people with a social emotional model, Neurocomputing 231 (2017) 47–54.

[22] W. Xie, C. Kang, See you, see me: Teenagers self-disclosure and regret of posting on social network site, Computers in Human Behavior 52 (2015) 398–407.

[23] A. Upadhyay, A. Chaudhari, S. Ghale, S. Pawar, et al., Detection and prevention measures for cyberbullying and online grooming, in: Inventive Systems and Control (ICISC), 2017 International Conference on, IEEE, 2017, pp. 1–4.

[24] M. Matsiola, C. Dimoulas, G. Kalliris, A. A. Veglis, Augmenting user interaction experience through embedded multimodal media agents in social networks, in: Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications, IGI Global, 2018, pp. 1972–1993.

[25] D. Rosaci, Cilios: Connectionist inductive learning and inter-ontology similarities for recommending information agents, Information systems 32 (6) (2007) 793–825.

[26] Q. Cao, M. J. Schniederjans, Agent-mediated architecture for reputation-based electronic tourism systems: A neural network approach, Information & Management 43 (5) (2006) 598–606.

[27] E. Franchi, A. Poggi, Multi-agent systems and social networks, in: Handbook of Research on Business Social Networking: Organizational, Managerial, and Technological Dimensions, IGI Global, 2012, pp. 84–97.

[28] F. Buccafurri, A. Comi, G. Lax, D. Rosaci, Experimenting with certified reputation in a competitive multi-agent scenario, IEEE Intelligent Systems 31 (1) (2015) 48–55.

[29] O. López-Ortega, I. Villar-Medina, A multi-agent system to construct production orders by employing an expert system and a neural network, Expert Systems with Applications 36 (2) (2009) 2937–2946.

[30] D. Rosaci, G. M. Sarnè, Multi-agent technology and ontologies to support personalization in b2c e-commerce, Electronic Commerce Research and Applications 13 (1) (2014) 13–23.

[31] C. Savaglio, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, G. Fortino, Agent-based internet of things: State-of-the-art and research challenges, Future Generation Computer Systems 102 (2020) 1038–1053.

[32] G. Fortino, F. Messina, D. Rosaci, G. M. Sarné, Using trust and local reputation for group formation in the cloud of things, Future Generation Computer Systems 89 (2018) 804–815.

[33] M. Alrubaian, M. Al-Qurishi, A. Alamri, M. Al-Rakhami, M. M. Hassan, G. Fortino, Credibility in online social networks: A survey, IEEE Access 7 (2018) 2828–2855.

[34] M. E. Gregori, J. P. Cámara, G. A. Bada, A jabber-based multi-agent system platform, in: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, ACM, 2006, pp. 1282–1284.

[35] P. D. O'Brien, R. C. Nicol, Fipa-towards a standard for software agents, BT Technology Journal 16 (3) (1998) 51–59.

[36] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015 (2014). URL `http://arxiv.org/abs/1412.6980`

[37] A. Mehrabian, Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament, Current Psychology 14 (4) (1996) 261–292.