MAS-based affective state analysis for user guiding in on-line social environments

DEPARTMENT OF INFORMATION
SYSTEMS AND COMPUTATION

Supervised by
Dr. Vicente Javier Julián Inglada
Dra. Ana María García Fornes

Dr. Agustín Rafael Espinosa Minguet

Guillem Aguado Sarrió

Valencia, January 2021

Abstract

In the present days, there is a strong and growing influence of on-line applications in our daily lives, and concretely Social Network Sites (SNSs) are one of the most used on-line social platforms that allow users to communicate and interact from different parts of the world every day. Since this interaction poses several risks, and also teenagers have characteristics that make them more vulnerable to certain risks, it is desirable that the system could be able to guide users when interacting on-line, to try and mitigate the probability of incurring one of those risks. This would in the end lead to a more satisfactory and safe experience for the users of such on-line platforms.

Recently, interest in artificial intelligence applications being able to perform sentiment analysis has risen. The uses of detecting the sentiment of users in on-line platforms or sites are variated and rewarding. Sentiment polarities can be used to perform opinion mining on people or products, and discover the inclinations and opinions of users on certain products (or certain features of them) to help marketing campaigns, and also on people such as politics, to discover the voting intention for example in electoral periods.

In this thesis, a Multi-Agent System (MAS) is presented, which integrates agents that perform different sentiment and stress analyses using text and keystroke dynamics data (using both unimodal and multi-modal analysis). The MAS uses the output of the analyzers for generating feedback for users and potentially avoids them from incurring risks and spreading comments in on-line social platforms that could lead to the spread of negative sentiment or high-stress levels. Moreover, the MAS incorporates parallelized analyses of different data types and feedback generation via the use of two different mechanisms. On the one hand, a rule-based advisor agent has been implemented, that generates feedback or guiding for users based on the output of the analyzers and a set of rules. On the other hand, a Case-Based Reasoning (CBR) module that uses not only the output of the different analyzers on the messages of the user interacting, but also context information from user interactions such as the topics being talked about or information about the previous states detected on messages written by people in the audience of the user.

Experiments with data from a private SNS generated in a laboratory with real people using the system in real-time, and also with data from Twitter.com have been performed to ascertain the efficacy of the different analyzers implemented and the CBR module on detecting states of the user that propagate more in the network, which leads to discovering which of the techniques is able to better prevent potential risks that users could face when interacting, and in which cases. Significant differences were found and the final version of the MAS incorporates the best-performing analyzer agents, a rule-based advisor agent, and a CBR module. In the end, this thesis aims to help intelligent systems developers to build systems that are able to detect the state of users interacting in on-line sites and prevent risks that they could face, leading to a more satisfactory and safe user experience.

Resumen

Recientemente, hay una fuerte y creciente influencia de aplicaciones en línea en nuestro día a día. Más concretamente las redes sociales se cuentan entre las plataformas en línea más usadas, que permiten a usuarios comunicarse e interactuar desde diferentes partes del mundo todos los días. Dado que estas interacciones conllevan diferentes riesgos, y además los adolescentes tienen características que los hacen más vulnerables a ciertos riesgos, es deseable que el sistema pueda guiar a los usuarios cuando se encuentren interactuando en línea, para intentar mitigar la probabilidad de que caigan en uno de estos riesgos. Esto conduce a una experiencia en línea más segura y satisfactoria para usuarios de este tipo de plataformas.

El interés en aplicaciones de inteligencia artificial capaces de realizar análisis de sentimientos ha crecido recientemente. Los usos de la detección automática de sentimiento de usuarios en plataformas en línea son variados y útiles. Se pueden usar polaridades de sentimiento para realizar minería de opiniones en personas o productos, y así descubrir las inclinaciones y opiniones de usuarios acerca de ciertos productos (o ciertas características de ellos), para ayudar en campañas de marketing, y también opiniones acerca de personas como políticos, para descubrir la intención de voto en un periodo electoral, por ejemplo.

En esta tesis, se presenta un Sistema Multi-Agente (SMA), el cual integra agentes que realizan diferentes análisis de sentimientos y de estrés usando texto y dinámicas de escritura (usando análisis unimodal y multimodal), y utiliza la respuesta de los analizadores para generar retroalimentación para los usuarios y potencialmente evitar que caigan en riesgos y difundan comentarios en plataformas sociales en línea que pudieran difundir polaridades de sentimiento negativas o niveles altos de estrés. El SMA implementa un análisis en paralelo de diferentes tipos de datos y generación de retroalimentación a través del uso de dos mecanismos diferentes. El primer mecanismo se trata de un agente que realiza generación de retroalimentación y guiado de usuarios basándose en un conjunto de reglas y la salida de los analizadores. El segundo mecanismo es un módulo de Razonamiento Basado en Casos (CBR) que usa no solo la salida de los analizadores en los mensajes del usuario interactuando para predecir si su interacción puede generar una futura repercusión negativa, sino también información de contexto de inter-

acciones de usuarios como son los tópicos sobre los que hablan o información sobre predicciones previas en mensajes escritos por la gente que conforma la audiencia del usuario.

Se han llevado a cabo experimentos con datos de una red social privada generada en laboratorio con gente real usando el sistema en tiempo real, y también con datos de Twitter.com para descubrir cuál es la eficacia de los diferentes analizadores implementados y del módulo CBR al detectar estados del usuario que se propagan más en la red social. Esto conlleva descubrir cuál de las técnicas puede prevenir mejor riesgos potenciales que los usuarios pueden sufrir cuando interactúan, y en qué casos. Se han encontrado diferencias estadísticamente significativas y la versión final del SMA incorpora los analizadores que mejores resultados obtuvieron, un agente asesor o guía basado en reglas y un módulo CBR. El trabajo de esta tesis pretende ayudar a futuros desarrolladores de sistemas inteligentes a crear sistemas que puedan detectar el estado de los usuarios interactuando en sitios en línea y prevenir riesgos que los usuarios pudiesen enfrentar. Esto propiciaría una experiencia de usuario más segura y satisfactoria.

Resum

Recentment, hi ha una forta i creixent influència d'aplicacions en línia en el nostre dia a dia, i concretament les xarxes socials es compten entre les plataformes en línia més utilitzades, que permeten a usuaris comunicar-se i interactuar des de diferents parts del món cada dia. Donat que aquestes interaccions comporten diferents riscos, i a més els adolescents tenen característiques que els fan més vulnerables a certs riscos, seria desitjable que el sistema poguera guiar als usuaris mentre es troben interactuant en línia, per així poder mitigar la probabilitat de caure en un d'aquests riscos. Açò comporta una experiència en línia més segura i satisfactòria per a usuaris d'aquest tipus de plataformes.

L'interés en aplicacions d'intel·ligència artificial capaces de realitzar anàlisi de sentiments ha crescut recentment. Els usos de la detecció automàtica de sentiments en usuaris en plataformes en línia són variats i útils. Es poden utilitzar polaritats de sentiment per a realitzar mineria d'opinions en persones o productes, i així descobrir les inclinacions i opinions d'usuaris sobre certs productes (o certes característiques d'ells), per a ajudar en campanyes de màrqueting, i també opinions sobre persones com polítics, per a descobrir la intenció de vot en un període electoral, per exemple.

En aquesta tesi, es presenta un Sistema Multi-Agent (SMA), que integra agents que implementen diferents anàlisis de sentiments i d'estrés utilitzant text i dinàmica d'escriptura (utilitzant anàlisi unimodal i multimodal), i utilitza la resposta dels analitzadors per a generar retroalimentació per als usuaris i potencialment evitar que caiguen en riscos i difonguen comentaris en plataformes socials en línia que pogueren difondre polaritats de sentiment negatives o nivells alts d'estrés. El SMA implementa una anàlisi en paral·lel de diferents tipus de dades i generació de retroalimentació a través de l'ús de dos mecanismes diferents. El primer mecanisme es tracta d'un agent que realitza generació de retroalimentació i guia d'usuaris basant-se en un conjunt de regles i l'eixida dels analitzadors. El segon mecanisme és un mòdul de Raonament Basat en Casos (CBR) que utilitza no solament l'eixida dels analitzadors en els missatges de l'usuari per a predir si la seua interacció pot generar una futura repercussió negativa, sinó també informació de context d'interaccions d'usuaris, com són els tòpics sobre els quals es parla o informació sobre prediccions prèvies en missatges escrits per la gent que forma

part de l'audiència de l'usuari.

S'han realitzat experiments amb dades d'una xarxa social privada generada al laboratori amb gent real utilitzant el sistema implementat en temps real, i també amb dades de Twitter.com per a descobrir quina és l'eficàcia dels diferents analitzadors implementats i del mòdul CBR en detectar estats de l'usuari que es propaguen més a la xarxa social. Açò comporta descobrir quina de les tècniques millor pot prevenir riscos potencials que els usuaris poden sofrir quan interactuen, i en quins casos. S'han trobat diferències estadísticament significatives i la versió final del SMA incorpora els analitzadors que millors resultats obtingueren, un agent assessor o guia basat en regles i un mòdul CBR. El treball d'aquesta tesi pretén ajudar a futurs dissenyadors de sistemes intel·ligents a crear sistemes que puguen detectar l'estat dels usuaris interactuant en llocs en línia i prevenir riscos que els usuaris poguessen enfrontar. Açò propiciaria una experiència d'usuari més segura i satisfactòria.

Agraïments

Després del llarg viatge que suposa el doctorat, és usual expressar agraïment a les persones que hi han contribuït a aquesta tasca d'una forma o altra:

Als meus pares Vicent i Carme, que han donat suport econòmic i han estat al meu costat des que tinc memòria, i al meu germà Eric per acompanyar-me també des de la infància i que em va acollir al seu pis mentre cursava el màster en intel·ligència artificial.

Als meus amics que han estat amb mi durant tants dies (i nits), i que han passat moltes coses amb mi (David, Bàrbarara, Andrés, Almudena, Javi, Jes, Miguel, Àlex, Quim, Ramon, Adri i Christian). Espere que puguem sortir moltes vegades més després de sobreviure a la pandèmia de SARS-CoV-2!

Als companys del departament de recerca on he treballat durant els quatre anys de duració d'aquesta tesi. Al nostre *jefazo* Vicent Botti, per descomptat als meus directors de tesi (Vicent Julián, Ana García i Agustí Espinosa), per donar-me l'oportunitat de crear aquesta tesi a la Universitat Politècnica de València. També als meus companys doctorands en el departament (o treballant a l'altre edifici), que també han estat amb mi durant aquests quatre anys (Jose, Javi, Christian i Elies).

Moltes gràcies a tots i totes!

Contents

CHAPTER 1

Introduction and objectives

1.1 Background

Nowadays, the influence of an increasing number of on-line applications in our daily life is growing. Between the most popular on-line sites, there are Social Network Sites (SNSs), where users can interact, talk to others, send messages, and perform different actions related to human on-line interaction. Therefore, people are more likely to suffer distress from risks or problems derived from this on-line interaction. Risks that can arise from the interaction of users in SNSs and potential negative outcomes from the interaction have been reviewed in [1]. Moreover, different kinds of risks are reviewed in [2] and [3]. Between the reviewed risks, there are content, contact, and commercial risks. Firstly, content risks are risks that appear when users can receive content that could potentially be harmful to them. As a matter of example, a user could receive pornography, violence-inciting or racist content, misinformation, that depending on the case of the user could lead to harmful undesired experiences or negative behavior derived from these experiences. An example result would be violent behavior or acting misguided by the misinformation and ending up in a negative situation as a consequence of it. Secondly, contact risks are the risks that appear when users met strangers and interact with them, potentially leading to harassment or privacy issues. Finally, commercial risks are related to getting spam or being asked for personal information, which are features from aggressive marketing campaigns. This could lead to undesired content being received such as advertisements and to other issues such as losing money from revealing the bank account data to strangers. Additionally, for the case of teenagers, it has been reported in [4] that they could be affected by several risks at SNSs and also that they have characteristics making them more vulnerable to such risks.

The risks mentioned can cause distress to users, potentially lead to harmful and unsatisfactory social experiences, and might also have negative consequences on user privacy and mental health or stress levels. Therefore, it might be desirable that a system could be of assistance to users while navigating and interacting on-line. Intelligent systems can have a wide set of applications, from guiding systems to efficiency maximization systems, and planning systems, citing a few examples. The former case can be applied to SNSs and other systems where there are interactions on-line between users, in an attempt to create a better, more satisfactory, and safe user experience. Furthermore, the decision-making process is present when interacting with other users in a SNS, and it can drive the interactions (e.g. posting content in a group of users, sending messages, or answering other messages). Decision making has been reported to be affected by the emotional state in [5]. In that work, the authors reviewed the effect of incidental moods, discrete emotions, integral affect, and regret on decision making. Incidental moods and discrete emotions are affective states not directly linked with the task being performed. They can originate from different sources, such as remembering about a person not involved in the current task being performed or a different scenario than the one in which the person currently is. Differently, integral affect refers to affective states originating from the task at hand. Finally, regret is a negative and conscious emotional reaction to self-decision making. Authors reported that incidental moods, discrete emotions, integral affect, and regret affect decision making. They reported that incidental moods affect decision making by altering the perception of the decision maker, and in the case of regret, it does so by acting as anticipating regret, as in thinking of the negative outcome before it happens.

Since decision making is affected by the emotional state of users, intelligent systems could use sentiment analysis to aid users to make decisions and guide them, aiming to prevent potential risks. Sentiment analysis is a line of work that aims at detecting the state of users navigating, but it is not the only aspect of a user that a system can track down for assessing their state. In [6], stress has been associated with an emotional state (high arousal and negative valence), and an algorithm for detecting stress and relaxation levels in text has been proposed. The algorithm is a variation of a sentiment strength detection algorithm called SentiStrength [7]. In this way, a system could be able to use both the information of the detection of general sentiment polarity and stress levels, when analyzing the state of users writing messages, aiming to assess the state of users in a more complete way, when trying to prevent negative repercussions on a SNS or other on-line environment. Additionally, the detection of sentiment and stress is not limited to text. Sentiment analysis has been successfully performed on text, audio, visual, multi-modal, and keystroke dynamics data, and stress analysis has been performed on text and keystroke data as well [8]. Moreover, when aiming to detect potential negative repercussions in an on-line site, a system is not limited to the information about the psychological state of users. Other sources of information related to the context of interactions, such as the history of polarities and stress levels of the user of past interactions, the history of the audience where the message is going to be posted, or the topic of messages are examples of context features that the system could use to improve the feedback for avoiding potentially negative outcomes in on-line social environments.

A system could use different strategies for both detecting the state of the user and for guiding and giving feedback to users in SNSs or other social environments. One way of using the information would be to select one analyzer and utilize it to compute the state of users based only on the output of the analysis, for example, sentiment analysis on text data of the messages written by users. Nevertheless, this approach would be lacking the ability to detect other aspects of the user state and context of the conversations being hold, or information about the users interacting. Another possibility would be to perform different analyses and then use a kind of fusion analysis on the data. According to [9], there are three main approaches in the literature for multi-modal fusion sentiment analysis, which are categorized by type of fusion (feature, decision, and hybrid). Firstly, feature level fusion fuses the information creating new features that have information from different data sources. Secondly, decision level fusion fuses different modalities in the semantic space. Finally, hybrid level fusion combines both previous approaches. Moreover, data fusion methods could also be combined for performing a fusion analysis of sentiment and stress as an example [10]. Following, the output of this analysis could be used to detect the state of users and then give feedback to them as in the previous case. Another possibility would be creating different analyzers, performing unimodal and multi-modal analyses, and then analyze which one works best in real-life scenarios to predict states of the user which could lead to generating more useful feedback. Finally, if other sources of information are to be taken into account by the system, and not only the state of users when generating feedback and aiming to prevent negative repercussions in the system, a different strategy that uses the multiple sources of information for solving this problem could be used. Case-Based Reasoning (CBR) is one method for addressing this task. In CBR systems, a reasoner searches through a base of cases of previous situations already seen by the system and for which it has given a solution, and reuses the similar ones to new cases for solving new problems [11]. Cases are constructed using different features that can have different characteristics to model the situation of the system, and then a CBR module computes similarities between new cases and previous ones to reuse previous solutions, and also updates the case base and adds new cases for adapting to the ever-changing situation of a real-life scenario. Therefore, a system could compute the state of the user using one or more analyzers (unimodal, multi-modal, or both), record other information such as topics of the messages (political, religious, or any topics that could be detected in the text), the users that are in the audience of a given message, and other information that could be relevant, and create a case with these features and a solution that could be a potential outcome in the system (e.g. a negative repercussion). Then, the case could be used to search in a case base previous cases of other interactions between users, and give a prediction about what could happen in the system, for later creating feedback aimed to prevent potential issues that could arise from the interaction between users.

To sum up, intelligent systems can be used to detect the state of users and other relevant information such as audiences and topics of messages, and then use this data in different ways to predict potential negative outcomes in an on-line social environment that could arise from the interaction between users. The predictions could be used to generate feedback to users, attempting to prevent issues and risks that users face when interacting, thus creating a more satisfactory and safe on-line experience. This is a task that grows in importance in the present days, as more users get immersed in the on-line scenario as an important part of their daily lives, and the variety of on-line applications and sites grows in numbers as well.

1.2 Objectives and scope

In this thesis, the main aim is to develop intelligent systems able to detect different aspects of the state of users interacting in on-line sites and information relevant from the interactions, integrate this data together, and use it to guide users and prevent potential risks or negative situations in on-line social environments. Concretely, the present thesis focuses on the following objectives or goals:

- Review of the existing literature in detection of the affective states as in sentiment and stress analysis and combined approaches with other technologies. Review of the state-of-art of user privacy and guiding in SNSs, using affective states or not. Identification of advantages and shortcomings of previous approaches.

- Specification of different components of a user-guiding and privacy-preserving system that analyzes the state of users interacting and generates feedback to prevent potential risks and lead to a more satisfactory and safe experience.

- Research of the technologies available for performing sentiment and stress analysis. Determination of which ones could be more useful for a system that guides users through their sentiment and stress levels. Research of the techniques that could be used to employ the different information available to a system from a SNS for guiding and recommending users.

- Development of a Multi-Agent System (MAS) that is able to communicate with a SNS or other on-line social platform and obtain information about messages and users, analyze the information, store it, and give feedback in real-time. The proposed MAS parallelizes different tasks using agents.

- Development of different unimodal and multi-modal sentiment and stress analysis models, and integration into agents of the MAS for guiding users.

- Development of an advisor agent that integrates the output of the analyzers to create feedback, using a set of rules.

- Development of a CBR module that uses information about the context of interactions in addition to the output of data analyzers. This context information is such as the topic being talked about or the history of polarization of previous messages from users and audience of messages. The CBR module uses the information for generating feedback to users and guide or recommend them.

- Integration of the different developed tools into the MAS for guiding users. Validation of the proposed techniques for assessing the capacity of prevention of potential risks from user interactions in real-time, and with real users on a SNS.

1.3 Structure of the Thesis

The thesis is structured in chapters as follows:

- **Chapter 1. Introduction and objectives:** In this chapter, motivations, background, and objectives of the present thesis are presented. The contributions created in the context of the thesis, research projects associated, and structure of the thesis are also listed.

- **Chapter 2. Review on MAS-based sentiment and stress analysis and user guiding:** Reviews works in the line of prevention of risks that can arise from social interaction in on-line environments, focusing on works using MAS-based technologies. For this purpose, both works in affective state detection and works that use MAS-based technologies for user recommendation and guiding are analyzed. Potential unexplored future lines of work are presented, based on the analysis of previous approaches in affective state detection and risk prevention and recommendation, which fill the gaps in the existing literature.

- **Chapter 3. MAS for sentiment, stress, and combined analysis on text data:** Presents a MAS integrating agents that perform sentiment, stress, and a combined analysis of both sentiment and stress on text data, using Bayesian classifiers. The MAS proposed generates feedback for users in real-time when they are interacting in on-line social environments, by means of analyzing the text messages for detecting the affective states of users, and use this information to warn them in case the state detected is negative according to the combined analysis. Experiments with data from the popular SNS Twitter.com and the proposed analyzers are shown. Those experiments aim to discover which analysis method is able to detect a state of the user that propagates more in the network, and thus being able to detect negative repercussions better (as in negative sentiment polarities or high-stress levels spreading through users in the network).

- **Chapter 4. Sentiment, stress and combined analyses using ANNs:** Exposes a new version of the MAS, in which agents performing sentiment and stress analysis on text using Artificial Neural Networks (ANNs) are implemented, and a new version of combined analysis proposed. Experiments with data from Twitter.com and from a private SNS called Pesedia [12] in a real-time scenario with users in a laboratory are conducted, for assessing the performance of the analyzers proposed and the effectiveness of the MAS in real-life scenarios, respectively. Surveys are also passed to users for understanding how they felt about the system and for knowing if they thought that problems could arise from interaction, and also if they like the idea of a system warning them about those.

- **Chapter 5. Keystroke dynamics analysis for sentiment and stress detection:** Presents new agents performing sentiment and stress analysis on keystroke dynamics data, which were integrated into the MAS. A variety of combined analyzers, combining sentiment and stress analysis on text, on keystroke data, and on both (multi-modal) are proposed and tested using data from Pesedia. In the experiments, the best-performing analyses are identified (for the task of detecting states of the user that propagate more in the on-line social environments). Additionally, a new agent using a set of rules and the best performing analyses is proposed, for allowing the MAS to give different feedback in distinct situations of the system (e.g. if the tokenizer for text-based sentiment analysis does not detect valid tokens in the text, the combined analyzer using keystroke data is used).

- **Chapter 6. CBR module using data analysis and context information:** Shows the proposal of a CBR module integrating the output of sentiment and stress analysis on text data and the same on keystroke dynamics data with context information from the interactions such as the topics detected in text messages. The CBR module is tested with Pesedia data for determining the error rate of the module when populating the case base with different configurations of the case features and different update intervals. Additionally, experiments comparing the analyzers on text and keystroke data with the CBR module are performed with Pesedia data again for testing the capacity to detect a state of users that propagated more in the network.

- **Chapter 7. Discussion:** In this chapter, a final discussion of results obtained in the different chapters discussed in this document is performed, general conclusions presented and future lines of work highlighted.

1.4 Publications list

In this section, the papers related to this thesis are shown. The different types of publications are separated. In following chapters, the contents published in [8], [13], [14], and [10], and under review in the journal Knowledge-Based Systems are elaborated. Those are in that same order presented at chapters 2, 3, 4, 5, and 6. They are also the journal articles exposed in this section following.

- Journals articles:

- G. Aguado, V. Julián, and A. Garcia-Fornes. *Towards aiding decision-making in social networks by using sentiment and stress combined analysis* **Information** Vol. 9, N. 5, pp. 107. (2018) DOI: `https://doi.org/10.3390/info9050107` **Impact Factor (SJR): 0.353. (Q3)**
- G. Aguado, V. Julián, A. Garcia-Fornes, and A. Espinosa. *A Multi-Agent System for guiding users in on-line social environments.* **Engineering Applications of Artificial Intelligence** Vol. 94, pp. 103740. (2020) DOI: `https://doi.org/10.1016/j.engappai.2020.103740` **Impact Factor (JCR): 4.201. (Q1)**
- G. Aguado, V. Julián, A. Garcia-Fornes, and A. Espinosa. *Using Keystroke Dynamics in a Multi-Agent System for User Guiding in Online Social Networks.* **Applied Sciences** Vol. 10, N. 11, pp. 3754. (2020) DOI: `https://doi.org/10.3390/app10113754` **Impact Factor (JCR): 2.474. (Q2)**
- G. Aguado, V. Julián, A. Garcia-Fornes, and A. Espinosa. *A Review on MAS-Based Sentiment and Stress Analysis User-Guiding and Risk-Prevention Systems in Social Network Analysis.* **Applied Sciences** Vol. 10, N. 19, pp. 6746. (2020) DOI: `https://doi.org/10.3390/app10196746` **Impact Factor (JCR): 2.474. (Q2)**
- Under review G. Aguado, V. Julián, A. Garcia-Fornes, and A. Espinosa. *A CBR for integrating sentiment and stress analysis for user guiding in social network sites* **Knowledge-Based Systems** (2020) **Impact Factor (JCR): 5.921. (Q1)**

- Conference articles:
  - G. Aguado, V. Julián, and A. Garcia-Fornes. *Multi-agent System for Privacy Protection Through User Emotions in Social Networks.* In: Bajo J. et al. (eds) Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems. PAAMS 2017. Communications in Computer and Information Science, vol 722, pp. 235-245. (2017) Print ISBN: 978-3-319-60284-4. Online ISBN: 978-3-319-60285-1. DOI: `https://doi.org/10.1007/978-3-319-60285-1_20`
  - G. Aguado, V. Julián, and A. Garcia-Fornes. *Rethinking Posts Through Emotion Awareness.* In: De la Prieta F. et al. (eds)

Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection - 15th International Conference, PAAMS 2017. Advances in Intelligent Systems and Computing, vol 619, pp. 262-263. (2017) Print ISBN: 978-3-319-61577-6. Online ISBN: 978-3-319-61578-3. DOI: `https://doi.org/10.1007/978-3-319-61578-3_32`

– G. Aguado, V. Julián, and A. Garcia-Fornes. *Analyzing the Repercussions of the Actions Based on the Emotional State in Social Networks.* In: Belardinelli F., Argente E. (eds) Multi-Agent Systems and Agreement Technologies. EUMAS 2017, AT 2017. Lecture Notes in Computer Science, vol 10767, pp. 523-537. (2017) Print ISBN: 978-3-030-01712-5. Online ISBN: 978-3-030-01713-2. DOI: `https://doi.org/10.1007/978-3-030-01713-2_37`

1.5 Research projects

This thesis was funded by the following research projects:

- ***Privacy in Social Educational Environments during Childhood and Adolescence (PESEDIA):***
  - Funder entity: Ministerio de Economia y Empresa (TIN2014-55206-R).
  - Lead Applicant: A. García-Fornes, and A. Espinosa.
  - Duration: 2015 - 2017.
- ***Intelligent Agents for Privacy Advice in Social Networks (AI4PRI):***
  - Funder entity: Ministerio de Economia y Empresa (TIN2017-89156-R).
  - Lead Applicant: E. Argente, and A. García-Fornes.
  - Duration: 2018 - 2020.

CHAPTER 2

Review on MAS-based sentiment and stress analysis and user guiding

2.1 Introduction

Works on sentiment analysis using text, audio, visual and physiological signals are reviewed in [9], including multi-modal sentiment analysis, which combines different data sources. Applications of sentiment analysis are also commented and future lines of work are highlighted. To the best of our knowledge, there is a lack of a review in user risk prevention when navigating on-line social environments, therefore in this chapter, works that address this line of work are reviewed, and works that use Multi-Agent System (MAS) based user recommendation are highlighted. This leads to also review works not only in sentiment analysis but in the lines of automatic user state detection, concretely the cases of sentiment analysis and stress analysis, so the state-of-art techniques used for detection can be later linked to prevention approaches in sections 5 and 6. In those sections, conclusions about how current technologies help prevent issues in on-line social platforms are drawn and potential new lines of work elaborated. Therefore, the aim of this chapter is to review the current state-of-art works in risk prevention and recommendation in on-line social environments using MAS-based approaches, and also review the literature in sentiment and stress detection, for being able to link them and to emphasize potential new lines of work, unexplored at the moment. Consequently, the present survey might help future researchers and developers to build on-line social platforms that could guide the users and prevent them from suffering negative consequences of interacting, thus leading to a more satisfactory and safe social experience. The survey was published in [8].

The rest of the chapter is structured as follows. Section 2 presents the main topics of this literature review. Section 3 reviews a series of recent works in the lines of sentiment and stress analysis. Section 4 reviews works in risk prevention in Social Network Sites (SNSs), and usage of MAS technologies for user guidance and recommendation. Section 5 makes an overview of the cases of study of automatic user state detection and gives insight on how could risk prevention and user guidance be addressed by detecting the user state in SNSs. Finally in section 6 future potential lines of work are extracted and developed and conclusions are extracted.

2.2 Problem statement

The present chapter aims to review the current literature on three different topics and to link them together, for extracting potential future lines of work in risk prevention in on-line social environments. The three topics are user state detection, risk prevention, and recommendation.

- User state detection: refers to the automatic detection of an aspect of the user state by the system. Has many variations, but since in this work we aim to address risk prevention we focus on the detection of the sentiment and stress levels of users, which are sentiment analysis and stress analysis. Those two techniques address the problem of detecting automatically the sentiment and stress level of users by employing different techniques (e.g. machine learning, natural language processing (NLP)) on different data sources (e.g. text, audio, images), either using one data modality or multiple.

- Risk prevention: addresses the prevention of risks that users on a system can suffer. In our case, we focus on the prevention of risks that users can suffer while navigating on-line social environments, such as SNSs. It can be performed by employing user state detection and applying feedback to users when necessary or giving recommendations to them, which is the focus of the present survey, but it can also be addressed by performing analysis of relations between users and warning users about dangerous people, as an example.

- Recommendation: encompasses the techniques used by a system to give recommendations about different matters to users (e.g. what to buy, when to invest, whom to trust). User state detection can be used by the system to perform recommendations, and recommendations can in turn be used to prevent risks that users could be exposed to in a system.

In the following two sections and for being able to later draw conclusions of how the existing state-of-art technologies can help risk prevention and user guidance in SNSs, and elaborate potential new lines of work, we will review works related to the user-guiding process. Firstly, we start reviewing works on user automatic sentiment polarity and stress level detection. We also review works in Case-Based Reasoning (CBR) based sentiment analysis. Later we review works in risk prevention, and MAS-based recommendation and guiding systems.

2.3 Detection approaches review

In the following subsections approaches on detection of the state of the user by analyzing different sources of data and fusion techniques will be reviewed. Performance of the reviewed approaches and techniques used are summarized in tables 2.1 and 2.2, respectively. Datasets used, their characteristics, and the partitions used for training and testing are summarized in tables 2.3 and 2.4 (these two later tables will be referred to as the 'dataset tables' in the following text of this chapter). From the dataset tables, it can be seen that annotated customer reviews about products in e-commerce websites such as Amazon are very useful to construct datasets for sentiment analysis models. Annotated reviews extracted from various sites not related to e-commerce can also be useful for this purpose, and they can be found from very variated sources (e.g. the Internet Movie Database, epinions.com, CitySearch.com). Moreover, in the dataset tables it is shown that when building a dataset, researchers can make use of data streaming services from SNSs, such as Twitter.com, or download videos and images from on-line platforms that allow users to share images and videos (e.g. Youtube, Flickr). Then, for labeling the data researchers can make use of labeling services such as the one offered by Amazon (Amazon Mechanical Turk or AMT). If there is no need for very large datasets, controlled laboratory experiments with a set of people to generate a dataset are always an option, or reusing existing datasets, as is shown as well in the dataset tables. Additionally, in the dataset tables can be seen that specific requirements such as people reacting to a specific set of emotions, or collecting data under certain conditions of physical and cognitive stress might require to conduct a laboratory experiment, since those might be hard to find and reuse. Stress-related data can be collected on-line like in the case of sentiment analysis datasets, using, for example, SNS data.

Firstly, we review sentiment analysis on text data, which is one of the most predominant and well-established lines of work in recent years regarding data analysis for detecting the state of the user. Following we review visual sentiment analysis, a more recent line of work on sentiment analysis that gives a new approach to the problem of detecting sentiment polarity of the users. In the same way, we later review works on another approach to the sentiment analysis task, which is sentiment analysis on audio data. We follow with a review of works on multi-modal fusion sentiment analysis, which refers to works that address sentiment analysis with a combination of techniques, using analysis of text and audio data and other sources of data. Also in these works, different levels of fusion are investigated. Next, we continue reviewing sentiment analysis works that use CBR technologies for their approaches. Finally, we finish with a review of works on stress analysis and works that use keystroke dynamics data to detect sentiment and stress. For covering the works most relevant to the aim of this review, we define an inclusion/exclusion criteria as follows:

- Is it relevant to the works being reviewed in a given section? (e.g. a work included in the sentiment analysis on audio data subsection has to focus on sentiment analysis techniques applied to audio).

- The work uses a technique or techniques to address the problem which are different from the ones used in other works reviewed previously in the same section? (e.g. dictionary-based methods are different from machine learning models for sentiment classification in sentiment analysis). There can be works using the same technique as another reviewed work if the problem addressed by the paper is different (e.g. one work addresses emotion detection using big data, which is different from other work performing emotion detection on stored data or on single users).

- Does the work provide experiments and data that gives insight on the usefulness of the used technique to address the problem at its focus? (e.g. accuracy, precision, and recall of the proposed technique or method on the addressed problem, or data on the significance of an effect, for example, data on the significance of the effect of emotion on keystroke latency).

For searching for works, different databases were used, which were Google Scholar and Web Of Science. An intensive search was performed on both of them for finding relevant works for the aim of this literature review.

2.3.1 Sentiment analysis on text

Sentiment analysis can be applied to different kinds of media. In this section, we will review state-of-art works in the line of applying sentiment analysis to texts and using distinct techniques. Sentiment analysis on texts has been assessed with four well-differentiated techniques in the literature, which are document-level sentiment analysis, sentence-level sentiment analysis, aspect-based sentiment analysis, and comparative sentiment analysis [36]. Starting from sentiment detected in an entire document, to sentiment in a sentence, and finally in an aspect, which is a sequence of words representing an aspect in the text (e.g. the government, love, group of people). In other words, depending on the level of fine-grained analysis that we want to perform we will be choosing one or another. Comparative sentiment analysis does not apply to this concept, it is an exception to the techniques of sentiment analysis on text mentioned in this paragraph, where we use comparative words as the elements of the model with a sentiment polarity associated. The model is trained so we can learn which ones are the preferred entities using comparative sentences [36].

Document-level sentiment analysis has two important issues. Having to give an aggregate value of sentiment for an entire document, and that it may, and potentially will contain a variated set of polarities in different sections of it. For these reasons, researchers developed more fine-grained levels of sentiment analysis. Sentence-level is easier but still could potentially find more than one polarity in the same sentence, which may lead to a conflict when trying to generate an aggregated sentiment for the sentence. Thus, aspect-level sentiment analysis was created, which focuses on concrete aspects or entities and tries to give as output a sentiment polarity associated with them. In [15] sentence-level sentiment analysis based on a sentiment lexicon and sentence syntactic structures is performed on Chinese texts, and further used to calculate the aggregated sentiment of the document where the sentences are used to compute a weighted sum of the polarity of each sentence in the document, considering the importance of each sentence. This work, even when addresses sentence-level and document-level sentiment analysis using sentences as atomic units for semantic analysis, is unable to perform a more fine-grained analysis, which is achieved with aspect-based sentiment analysis.

For the case of aspect-based sentiment analysis, there are two main problems to address, which are aspect detection and sentiment classification. Aspect detection is the detection of aspects from the training data for the model and sentiment classification is the actual sentiment analysis on the aspects, to give a sentiment label to them. There are several approaches for solving both of those problems, and also hybrid approaches that try to assess them at the same time [37]. For the case of aspect detection, frequency-based methods use the terms found in the training corpus with higher frequency as aspects for the aspect set of the sentiment analysis model [16]. Generative models are used for detecting aspects as well, in [17] Conditional Random Fields (CRFs) with a variated set of features are used. The frequency-based method generates aspects using the most frequent terms, while generative models use string tokens and their associated features for this matter. The frequency-based method presented in [16] has slightly less precision but significantly more recall than the one achieved in [17]. Sentiment classification is addressed with dictionary-based methods in [16], where a dictionary is generated by propagating the sentiment of a set of seed words through WordNet synonym/antonym graph, counting only adjectives as sentiment words. Then the dictionary is used for calculating sentence-level sentiment analysis using majority voting of the adjectives detected in a sentence and found in the dictionary, with some refinements such as flipping the polarity of negated adjectives or multiple polarities finding when the number of positive and negative words found in a sentence is the same. Supervised and unsupervised machine learning methods are also used for sentiment classification. In [18] a Support Vector Regression (SVR) model is used to find the sentiment score of aspects as a real number in the interval zero to five. In [19] each aspect is used to find phrases that could contain sentiment, and then a label is assigned to them using a method from computer vision called relaxation labeling. Dictionary-based approaches use an existing set of labeled terms, while supervised and unsupervised machine learning options use different machine learning techniques to assign labels to new terms. The reported precision was higher for machine learning approaches in this case than the dictionary-based method, but the recall was similar, although it reached higher values in some cases for machine learning approaches. Hybrid approaches try to detect aspects and assign sentiment polarities to them at the same time. In [20] a syntax-based method was used to extract other aspects from words associated with a sentiment, exploiting semantic relations. Generative models are also used, in [21] CRF are used to relate sentiments to aspects, extracting information from the relations between words. Both of the hybrid methods reviewed shown a high precision compared to other reviewed methods, but only the CRF approach presented in [21] reached high recall.

Applying the criteria for inclusion/exclusion of works relevant to this review, the work in [15] was selected for applying sentiment analysis at sentence-level and document-level, while others were selected for illustrating aspect-level sentiment analysis, using different techniques, and thus giving a global insight on the state-of-art in these lines of work. In [17] a frequency-based method is presented for aspect detection, while in [17] CRFs are used. Those two techniques are different as one uses frequent terms, while the other extracts features using a model and features related to text tokens. For the case of sentiment classification, different kinds of techniques are presented. In [16] dictionary-based methods are used, while in [18] supervised machine learning models are presented, and in [19] an unsupervised method is used. Finally, two different works using hybrid techniques for aspect detection and sentiment classification are reviewed. A syntax-based method is presented in [20], while CRFs are used in [21].

2.3.2 Visual sentiment analysis

In general, there are three main state-of-art approaches to visual sentiment analysis [38], which are mid-level sentiment ontology, deep sentiment prediction, and multi-modal sentiment prediction.

Mid-level sentiment ontology provides a set of mid-level features for visual sentiment analysis. Borth and Ji [22] proposed a mid-level feature named adjective-noun pairs (ANPs), which are related to sentiments and a visual sentiment ontology (VSO). These ANPs with sentiment are extracted from a set of annotated images, then they are used to train VSO detectors, and a set of detectors are used to construct SentiBank, which is finally used for sentiment prediction on images. Deep sentiment prediction uses deep learning models to predict sentiment on images. Yu et al [23] proposed a progressive Convolutional Neural Network (CNN) that used a selection of images from the training set, according to the output of the trained model on them to fine-tune the training, and also address domain transfer by using a set of manually labeled Twitter.com images to fine-tune a previously trained CNN model. Finally, multi-modal sentiment prediction refers to creating classifiers that use different kinds of data to predict a sentiment, such as text and images. A multi-modal correlation model based on Markov Random Fields (MRFs) was proposed on [39]. Multi-modal features are extracted, with ANPs as image features, words as text features, and symbols as emoticons features. A MRF model is created and decomposed into 4 subgraphs (three single graphs and one correlation graph). The single graph denotes the contribution of each modality to the sentiment, and the correlation graph denotes the correlation contribution. According to the correlation of each modality, the final model is graphed and then learned. Following the inclusion/exclusion criteria, in this section the three different approaches used in the literature for visual sentiment analysis are presented in three different works. The three different strategies or techniques have clear differences, either use mid-level features that help predict an emotion from images, such as ANP that are text concepts related to the image (adjective-noun pairs), deep learning models, or a multi-modal model that analyzes both text and images to predict emotion. Deep models were reported to have reached a better accuracy overall but slightly less maximum accuracy than mid-level sentiment ontology prediction in the reviewed works.

2.3.3 Sentiment analysis on audio data

Sentiment analysis has been successfully performed using only audio data. In [24] an Automatic Speech Recognition (ASR) system is used to convert youtube videos to transcribed text, and then a Part Of Speech (POS) tagger based feature extraction technique to identify useful sentiment features in the text, to later be classified into positive or negative sentiment polarity by a Maximum Entropy (ME) based sentiment classification model; a method is proposed in [25] to perform speech emotion recognition. The proposed method performs emotion recognition by employing Vowel-Like Regions (VLRs) and Non-Vowel-Like Regions (non-VLRs), and by choosing the features of either VLRs or non-VLRs for each emotion; a sparse autoencoder based feature transfer learning method is proposed in [26] that uses a single-layer autoencoder to find a common structure in small target data and then applies it to reconstruct source data for performing knowledge transfer from source data into target task. The authors used the reconstructed data to build a speech emotion recognition engine for a real-life task and performed experiments with six publicly available corpora. The experiments showed that the proposed algorithm enhances emotion classification accuracy of the speech emotion recognition engine significantly; sentiment analysis on short spoken reviews was performed in [27], where authors collected manually a set of user reviews and extracted acoustic features from the spoken reviews using the openEAR/openSMILE toolkit. Various algorithms were used, which were logistic regression, AdaBoost, a C4.5 decision tree, and a Support Vector Machine (SVM) classifier with a radial basis function (RBF) kernel. The best performing algorithm was AdaBoost with speaker-dependent features after applying manual feature selection, reaching an accuracy of 72.9%. The four works reviewed in this section give an overview of the different possibilities for emotion recognition on audio, having a classic ME model applied to POS features from transcripts, a work using different features and selecting concrete features for each emotion, a deep learning approach that leverages classic model detection, and a work using different algorithms. Therefore, we followed the inclusion/exclusion criteria in this section. The reported results shown a better accuracy reached by the work in [25], using different features for each emotion, although the best overall results of accuracy are achieved by the work performed in [27], where different algorithms are used.

2.3.4 Multi-modal fusion sentiment analysis

Regarding multi-modal fusion sentiment analysis, there are three main approaches in the literature, which are categorized by type of fusion (feature, decision, and hybrid) [9]. Feature level fusion fuses information creating features that have information of different data sources; decision level fusion fuses different modalities in the semantic space; hybrid level fusion combines both feature and decision level fusion.

A method for estimating spontaneously expressed emotions in audio-visual data was developed in [28]. Support Vector Regression and decision level fusion achieved an average performance gain of 17.6% and 12.7% over the individual audio and visual emotion recognition methods respectively. In [29], a feature-level fusion approach for recognizing emotion from video and physiological data was developed. The authors perform emotion recognition on the valence-arousal emotional space using Hidden Markov models (HMMs). The best recognition accuracies reported are 85.63% for arousal and 83.98% for valence. A comparison with the proposed feature-level fusion approach against a decision-level fusion and non-fusion approaches was performed on the same DEAP database, showing that significant improvements in accuracy obtained by the feature-level fusion approach were observed. A hybrid output-associative fusion method for emotion prediction in the valence and arousal space was proposed in [30]. The authors used facial expression, shoulder gesture, and audio cues, and used Bidirectional Long Short-Term Memory Neural Networks (BLSTM-NNs) and Support Vector Regression models for performing sentiment classification. The authors claim that their hybrid method outperforms results from predicting either valence or arousal alone (both for feature-level and model level fusion). A model that combines emotion aware big data and cloud technology with 5G is proposed in [31]. The authors claim that the proposed approach achieves 83.10% emotion recognition accuracy. The different alternatives of multi-modal sentiment analysis, which are presented in [9] are used in the works presented in this section, achieving different results. The decision-level fusion method presented in [28] reported a higher overall correlation between the estimates and reference values of emotion than the hybrid output-associative fusion method presented in [30], and the feature-level fusion method presented in [29] not only achieved higher accuracy than the unimodal and decision-level fusion approaches compared in the same work, but also achieved higher accuracy than the big data approach presented in [31]. These works were selected to give insight on the different alternatives of multi-modal sentiment analysis according to the inclusion/exclusion criteria, since each one uses one alternative, except the work in [31] which follows a different aim, that is applying emotion recognition combined with big data technology.

2.3.5 CBR for sentiment analysis

The CBR approach has successfully been applied in the past to predict sentiment. In [32] explicit customer needs are extracted from customer reviews of products, by means of performing sentiment analysis on them using fuzzy SVMs. Then, a CBR module constructs a case on its case base according to the ordinary use cases of products detected in the previous step. When extraordinary use cases are detected, the CBR module searches for the most similar ordinary case in the case base, and then elicits the extraordinary customer needs in the extraordinary use case based on the ordinary case employing substitution, rule-based adaptation, and design engineer evaluation of the adapted extraordinary cases. In [33] sentiment classification is addressed using a CBR-based approach. First, the case base is populated using labeled customer reviews and five different sentiment lexicons. If a document is correctly classified by at least one lexicon, a case is created containing document statistics and writing style of the review that generated it, and also the case solution which is the information about which sentiment lexicons generated a successful prediction on the review associated to the case. Prediction on new reviews is performed by retrieving the k most similar cases (1, 2, or 3 in the reported experiments), and querying the lexicons of the retrieved solutions for sentiment information on the terms of the new review. Domain ontology with natural language processing techniques are combined in [34] to perform sentiment analysis. Case-based reasoning is also used to learn from past sentiment polarizations. The authors claim that the accuracy obtained by the proposed model overcomes standard statistical approaches. Different ways of using a CBR-based approach for sentiment analysis are shown in the works presented in this section. While in [32] CBR is used for generating extraordinary cases (cases of use of products not frequently found), based on ordinary or frequent cases, in [33] the CBR approach is combined with sentiment lexicons to form cases of document statistics and writing style associated to lexicons that correctly classified the document that generated the case. Finally, in [34] CBR is used to learn from past sentiment polarizations while using domain ontology combined with NLP techniques for performing sentiment analysis. In the results can be observed that the later work outperforms the CBR approach for generating extraordinary use cases in precision and recall, and also outperforms the lexicon CBR approach in terms of accuracy. The inclusion/exclusion criteria was applied in this section selecting works that applied a CBR-based approach for sentiment analysis in different ways.

2.3.6 Stress analysis and keystroke dynamics

Stress analysis has been addressed in the literature using different sources of information, like in the case of sentiment analysis. In [6] text is used as the source of information for an algorithm that employs a lexicon of stress-related terms for calculating the stress score of a sentence, based on the score of the highest stress term found, with some rules that modify the base approach (e.g. spelling correction or negating stress words). Keystroke dynamics refers to the way un user types at a keyboard and includes different features such as timing features of key press and release and accuracy rate while typing. As an example of keystroke dynamics applied to stress analysis, in [40] authors show through a Wilcoxon Signed Rank test that several keystroke dynamics features reject the null hypothesis that stress and non-stress data does not have a significant difference at least at 90 % confidence level. Another example is [35], where authors used keystroke dynamics and linguistic features for the analysis of free text, and demonstrated that such techniques can be effectively used to detect cognitive and physical stress from free-text data. The authors claim that the accuracy of detection of cognitive stress was consistent with those obtained using affective computing methods, and that the accuracy for the detection of physical stress, while not being as high as the ones obtained for cognitive stress, still encourages further research. There are works that perform sentiment analysis using a model trained with keystroke dynamics data. In [41] the authors used IADS [42] sounds for inducing sentiments to a series of users and record their keystroke dynamics after hearing them. They showed that the effect of arousal on keystroke duration and keystroke latency was significant but the one on the accuracy rate of keyboard typing was not. The works performed at [40] and [41] demonstrate that there is an effect of stress and sentiment, respectively, on keystroke dynamics data, showing that models can be built to use this kind of data for performing sentiment and stress analysis. Keystroke dynamics data used in [35] to detect stress, and text data used in [6] for this same purpose can be compared in table 2.1, showing that the text-based method outperforms the keystroke method when using stress strength detection while allowing the matches to be +-1 stress level away of the label, while being outperformed when considering exact stress level matches.

Related to the inclusion/exclusion criteria, the works [40] and [41] are selected for the review because they perform an analysis on the significance of the effect of stress and sentiment on keystroke data, respectively. For reviewing a method for detecting stress, the works [6] and [40] were selected, since a method that addressed stress analysis on text data is presented in the former and one addressing stress analysis on keystroke data is shown in the later.

Table 2.1: Performance of the detection approaches

Reference	***Metrics***	***Values***
[15]	Accuracy, precision, recall and F1	0.6778 - 0.7961, 0.628 - 0.8905, 0.5931 - 0.9231 and 0.5549 - 0.8591
[16]	Precision and recall	0.56-0.79 and 0.67-0.80
[17]	Precision, recall and F1	Single-domain approach: 0.622 - 0.792, 0.414 - 0.661 and 0.497 - 0.702
		Cross-domain approach: 0.565 - 0.678, 0.273 - 0.435 and 0.36 - 0.518
[18]	Acuraccy, precision, recall and F1	Multi-aspect sentence labeling: 0.477 - 0.83, 0.126 - 0.969, 0.179 - 1 and 0.148 - 0.887
	L1 error, ρaspect, ρreview and MAP @10	Multi-aspect rating prediction with indirect supervision: 0.238 - 0.645, -0.149 - 0.715, 0.454 - 0.846 and 0.129 - 0.429
	L1 error	Supervised multi-aspect rating prediction: 0.554 - 1.071
[19]	Precision and recall	Explicit feature extraction task: 93-95% and 73-80%
		Finding word semantic orientation: 72-88% and 55-83%
		Extracting opinion phrases: 79% and 76%
		Extracting opinion phrase polarity: 86% and 89%
[20]	Precision and recall	0.75 - 0.955 and 0.2-0.286
[21]	Precision, recall and F1	0.761 - 0.918, 0.37 - 0.82, and 0.498 - 0.837
[22]	Accuracy	Using visual features: 0.49 - 0.83
		Using visual and text features: 0.48 - 0.88
[23]	Precision, recall, accuracy and F1	0.691 - 0.797, 0.729 - 0.905, 0.667 - 0.783 and 0.722 - 0.846
[24]	Accuracy	0.68 - 0.82
[25]	Accuracy	0.452 - 0.851
[26]	Unweighted Average Recall (UAR)	0.579 – 0.627
[27]	Accuracy (over automatic speech recognition output and human transcripts)	Using text and acoustic features: 0.825 and 0.81
		Using only text: 0.75 and 0.844
[28]	Mean linear error and correlation between estimates and the reference for valence, activation, and dominance	Acoustic emotion estimation: 0.13, 0.16, 0.14 and 0.53, 0.82, 0.78
		Visual emotion estimation (eyes region): 0.18, 0.19, 0.13 and 0.57, 0.58, 0.57
		Visual emotion estimation (lips region): 0.18, 0.19, 0.14 and 0.58, 0.62, 0.53
		Decision-level fusion acoustic and visual emotion estimation: 0.14, 0.12, 0.09 and 0.7, 0.84, 0.8
[29]	Accuracy for arousal and valence	Unimodal (Electroencephalogram): 0.65 - 0.7563 and 0.73 - 0.74
		Unimodal (Peripheral physiological signals): 0.6689 - 0.6905 and 0.6689 - 0.6905
		Feature-level fusion: 0.7844 - 0.8563 and 0.8279 - 0.8398
		Decision-level fusion: 0.66 - 0.73 and 0.5 - 0.64
[30]	Root Mean Squared Error (RMSE), Correlation Coefficient (COR) and Sign Agreement Metric (SAGR) for valence and arousal	Single-cue prediction: 0.17 - 0.22, 0.444 - 0.712, 0.648 - 0.841 and 0.24 - 0.29, 0.411 - 0.586, 0.681 - 0.764
		Support vector regression: 0.21 - 0.25, 0.146 - 0.551, 0.538 - 0.740 and 0.26 - 0.27, 0.388 - 0.419, 0.667 - 0.716
		Feature-level fusion: 0.19 - 0.21, 0.583 - 0.681, 0.733 - 0.856 and 0.24 - 0.28, 0.461 - 0.589, 0.685 - 0.763
		Model-level fusion: 0.16 - 0.19, 0.653 - 0.782, 0.830 - 0.892 and 0.22 - 0.26, 0.479 - 0.639, 0.637 - 0.8
		Output-associative fusion: 0.15 - 0.18, 0.664 - 0.796, 0.825 - 0.907 and 0.21 - 0.24, 0.536 - 0.642, 0.719 - 0.8
[31]	Accuracy and speed up factor (big data tools)	0.831 and 75.55
[32]	Precision and recall (sentiment analysis)	0.751 and 0.758
	Reduction of product attribute redundancy	0.41
[33]	Accuracy	0.62 - 0.7258
[34]	Accuracy, precision, recall and F1	0.85 - 0.91 0.85 - 0.899, 0.847 - 0.92 and 0.85 - 0.91
[6]	Exact matches with strength label, 1 level away of strength matches (+-1 of the label), correlation and MAD	Stress strength detection: 0.31 - 0.579, 0.791 - 0.939, 0.329 - 0.505 and 0.502 - 0.893
		Relaxation strength detection: 0.482 - 0.717, 0.916 - 0.963, 0.332 - 0.466 and 0.338 - 0.515
[35]	Accuracy	Detecting cognitive stress: 0.521 - 0.615 (raw data), 0.641 - 0.75 (normalized data)
		Detecting physical stress: 0.541 - 0.625 (raw data), 0.542 - 0.625 (normalized data)
	False Acceptance Rates (FARs) and False Rejection Rates (FRRs)	Detecting cognitive stress: 0.375 - 0.75 and 0.208 - 0.479 (raw data), 0.187 - 0.333 and 0.229 - 0.479 (normalized data)
		Detecting physical stress: 0.437 - 0.771 and 0.125 - 0.375 (raw data), 0.25 - 0.437 and 0.333 - 0.5 (normalized data)

Table 2.2: Techniques of detection approaches

Reference	***Technique***
[15]	Sentence-level sentiment analysis based on lexicon and syntactic structures. Document-level sentiment analysis using a weighted sum of the polarities of sentences within the document.
[16]	Part of Speech (POS) tagging. Frequent feature generation using association rule mining. Feature pruning as pruning meaningless and redundant features. Opinion words extraction, extracting words that are adjacent to frequent features and are adjectives that modify the feature. Infrequent feature identification using opinion words to find the nearest noun/noun phrase of the opinion word.
[17]	Conditional Random Fields (CRFs) for extracting aspects. Features: token string text, POS tag of the token text, label of existence of short dependency path (between the token and opinion expressions), word distance label (token appear in the closest word distance to an opinion expression or not), opinion sentence label (does the token appear in a sentence with opinion expressions?).
[18]	Multi-aspect sentence labeling: Latent Dirichlet Allocation (LDA), Multi-grain LDA (MG-LDA), Segmented Topic Model (STM) and Local LDA models weakly supervised using seed words, supervised Support Vector Machine (SVM) and a majority baseline that assigns the most common aspect label. For multi-aspect rating prediction with indirect supervision: LDA, MG-LDA, STM, and local LDA weakly supervised with seed words are used to label sentences with aspects, and a Support Vector Regression (SVR) model is trained with the combined vectors of each entity and their overall ratings. Supervised multi-aspect rating prediction: Perceptron Ranking (PRank) and linear SVR are used, trained with and without features derived from LDA, MG-LDA, STM, and local LDA, that do not make use of seed words, and trained with unigram baseline features.
[19]	Feature extraction: the algorithm extracts frequent noun phrases from parsed reviews. It also examines opinion phrases associated with explicit features in order to extract implicit properties. Finding opinion phrases: if there is an explicit feature in a sentence, the algorithm applies extraction rules to find opinion phrases. Each head word, together with its modifiers, is returned as a potential opinion phrase. Opinion phrases polarity detection: relaxation labeling.
[20]	POS tagging and shallow parser. Syntactic parsing and sentiment lexicon used for sentiment analysis and relating sentiment expressions to subjects.
[21]	CRFs. Joint extraction of opinions and object features.
[22]	Linear SVMs and Logistic Regression (LR) for learning Ajective Noun pairs (ANPs) detectors for visual sentiment analysis. Visual Sentiment Ontology (VSO) based on ANPs. SentiBank, visual concept detection library that can detect 1,200 ANPs in images using ANP detectors, based on VSO.
[23]	Progressive Convolutional Neural Network (PCNN) for image sentiment classification trained using weakly labeled data. Data from the output of the model is used to fine-tune it. Previously trained CNNs are fine-tuned with a small set of manually labeled images for addressing domain transfer.
[24]	Automatic Speech Recognition (ASR) for obtaining transcriptions of videos. POS tagging for extracting text-based sentiment features. A Maximum Entropy (ME) model with feature tuning for sentiment classification.
[25]	Algorithm for emotion classification using region switching between Vowel-like Regions (VLR) and non-VLRs from audio data.
[26]	Sparse autoencoder-based feature transfer learning method, using a single-layer autoencoder to find a common structure in small target data and then applying such structure to reconstruct source data.
[27]	ASR engine: The AT&T Watson speech recognizer was used to convert spoken review summaries to text. Linear interpolation of the three Katz's backoff language models. In the experiments, Ada-boost with acoustic features combined with a text-based prediction feature was used and compared to LR, SVM, and C4.5 decision tree.
[28]	Acoustic emotion estimation: SVR. Visual emotion estimation: SVR. Decision-level fusion: weighted linear combination of the acoustic and visual estimations for a given sentence using different weights for the estimation of valence, activation and dominance.
[29]	Hidden Markov Models (HMMs) using multi-modal feature sets. Electroencephalogram (EEG) from the central nervous system and three kinds of Peripheral physiological signals (PERI) from the peripheral nervous system (Respiration or RSP, Electromyogram or EMG, and Skin Temperature or TMP) are used. Fusion at feature level is performed, and at decision level employing six different strategies, classification is performed using feature fusion, decision fusion, and non-fusion models.
[30]	SVR and bidirectional Long Short-Term Memory Neural Networks (BLSTM-NN) both used for single-cue prediction of valence and arousal. BLSTM-NNs are used for feature-level and model-level fusion, as well as for output-associative fusion of different cues (facial Expressions, shoulder cues, and audio cues). Model-level fusion performs fusion of the output of BLSTM-NNs predicting valence or arousal, using different cues (one cue in each BLSTM-NN), and uses it as the input for another BLSTM-NN. Output-associative fusion fuses the output of BLSTM-NNs predicting valence and BLSTM-NNs predicting arousal using different cues (again one cue in each BLSTM-NN).
[31]	Audio-visual big data emotion recognition system, using Multi-directional Regression (MDR) features for speech and Weber Local Descriptor (WLD) features for face images. SVM classifiers are used for each modality and decision-level fusion using Bayesian sum rule.
[32]	Fuzzy SVMs for sentiment analysis on customer reviews of products. Case-based Reasoning (CBR) for generating ordinary and extraordinary use cases, from the sentiment labeled product attributes obtained from the SVMs.
[33]	CBR to compare text documents and different sentiment lexicons for sentiment classification associated to the cases.
[34]	Domain ontology and POS tagging for sentiment analysis, using CBR for reusing past cases of sentiment detection on text.
[6]	Algorithmic approach using a lexicon of stress and relaxation terms to detect stress and relaxation magnitude on text. The value predicted on a sentence is based on the score of the highest stress or relaxation term found within that sentence. Sentiment on a text with more than one sentence is computed as the highest value from any constituent sentence. Corrections such as negation of stress terms or spelling correction are applied.
[35]	Decision Tree (DT), SVM, k-Nearest Neighbor (kNN), AdaBoost, using DecisionStump as a base classifier, and ANN are used. DT was used to select features as input for the other methods.

Table 2.3: Datasets used and partitions for training and testing of the detection approaches

Reference	*Dataset or datasets*	*Partitions*
[15]	Euthanasia dataset: 851 Chinese articles on "euthanasia", manually labeled into 502 positive and 349 negative articles. AmazonCN dataset: 458,522 reviews from six categories (book, music, movie, electrical appliance, digital product, and camera), labeled according to Amazon user's five-star rating into 310,390 positive and 29,540 negative reviews.	Euthanasia dataset: Standard 10-fold cross-validation was performed. AmazonCN dataset: Up to 200 positive and 200 negative randomly selected reviews of each product category as the training dataset, and up to 500 positive and 500 negative randomly selected reviews of each product category as the test dataset.
[16]	Customer reviews from Amazon.com and C—net.com about five products (2 digital cameras, 1 DVD player, 1 mp3 player, and 1 cellular phone). 100 reviews for each product. A person extracted features manually for evaluation, resulting in 79, 96, 67, 57, and 49 manual features for Digital camera1, Digital camera2, Cellular phone, Mp3 player, and DVD player, respectively.	The data was used in the proposed system to perform feature extraction and compare it to the manually extracted features.
[17]	Four datasets annotated with individual opinion target instances on a sentence level. Movies: reviews for 20 movies from the Internet Movie Database (1829 documents containing 24555 sentences). Annotated with opinion target - opinion expression pairs. Web-services: reviews for two web-services collected from epinions.com (234 documents containing 6091 sentences). Cars: Reviews of cars (336 documents containing 10969 sentences). Cameras: blog postings regarding digital cameras (234 documents containing 6091 sentences). In order for datasets movies, web-services, cars and cameras: sentences with targets: 21.4%, 22.4%, 51.1% and 54.0%; sentences with opinions: 21.4%, 22.4%, 53.5% and 56.1%.	As development data for the CRF model, 29 documents from the movies dataset, 23 documents from the web-services dataset, and 15 documents from the cars and cameras datasets were used. 10-fold cross-validation in single-domain (single dataset) experiments. In the cross-domain experiments, the system is trained on the complete set of data from one or various datasets and tested on all the data of a dataset not used in training.
[18]	OpenTable: 73,495 reviews (29,596 after excluding excessively long and short reviews) and their associated overall, food, service, and ambiance aspect ratings for all restaurants in the New York/Tri-State area appearing on OpenTable.com. Not labeled; CitySearch: 652 restaurant reviews from CitySearch.com. Each sentence manually labeled with one of six aspects: food, service, ambiance, price, anecdotes, or miscellaneous; TripAdvisor: 66,512 hotel reviews. Each review labeled with overall rating and ratings for 7 aspects: value, room, location, cleanliness, check-in/front desk, service, and business services.	Multi-aspect sentence labeling: For evaluation, 1,490 singly-labeled sentences from the annotated portion of the CitySearch corpus were used. Inference is performed on all 652 documents of CitySearch; multi-aspect rating prediction with indirect supervision: OpenTable and TripAdvisor datasets sentences are labeled with aspects using weakly supervised topic models. All reviews for each entity (hotel or restaurant) are combined into a single review, aspect ratings are obtained by averaging the overall/aspect ratings for each combined review. 5-fold cross-validation is performed; supervised multi-aspect rating prediction: 5-fold cross-validation on subsets of the OpenTable and TripAdvisor data.
[19]	Two sets of 1307 reviews downloaded from tripadvisor.com for Hotels and amazon.com for Scanners. Two annotators labeled a set of 450 feature extractions from the algorithm as correct or incorrect. The annotators extracted explicit features from 800 review sentences (400 for each domain); Word semantic dataset: 13841 sentences and 538 previously extracted features; Opinion phrase dataset: 550 sentences containing previously extracted features. The sentences were annotated with opinion phrases corresponding to the known features and with opinion polarity.	Explicit feature extraction: the algorithm was evaluated on the two sets from TripAdvisor and amazon. Finding word semantic orientation: the algorithm was evaluated on the Word semantic dataset. Extracting pinion phrases and opinion phrase polarity detection: the algorithm was evaluated on the Opinion phrase dataset.
[20]	Benchmark Corpus: 175 samples of subject terms within context text. Contains 118 favorable sentiment samples and 58 unfavorable samples; Open Test Corpus: 2,000 samples related to camera reviews. Half the samples are labeled favorable or unfavorable and the other half neutral; 6,415 web pages with 16,862 subject references, 1,618 news articles with 5,600 subject references, 1,198 pharmaceutical web pages with 3,804 subject references.	The system was directly used with data from the datasets.
[21]	Movies dataset: 500 reviews about 5 movies. Contains 2207 sentences; Products dataset: 601 reviews about 4 products. Contains 2533 sentences. Both datasets are labeled manually by humans. Labels for object features, positive opinions, negative opinions, and the object feature-opinion pairs for all sentences are given.	Each dataset is split into 5 parts, and four are used for training while one for testing.
[22]	Flickr dataset: 150,034 images and videos with 3,138,795 tags; YouTube dataset: 166,342 images and videos with 3,079,526 tags; Amazon Mechanical Turk (AMT) experiment: randomly sampled images of 200 Adjective Noun Pair (ANP) concepts from the Flickr images, manually labeled by AMT crowdsource; Twitter Images dataset: Tweets containing images crawled using popular hashtags. Three labeling runs using AMT, namely image-based, text-based, and joint text-image based are performed. The dataset includes 470 positive tweets and 133 negative tweets over 21 hashtags; ArtPhotos dataset: ArtPhotos retrieved from DeviantArt.com. Contains 807 images from 8 emotion categories.	Training dataset with Flickr ANP labeled images: 80% of pseudo positive images of each ANP and twice as many negative images. Test datasets (full and reduced test sets): both use 20% of pseudo positive samples of a given ANP as positive test samples. The full test set includes 20% pseudo positive samples from each of the other ANPs (except those with the same adjective or noun) as negative samples. The reduced test set contains twice as many negative samples for each ANP as the positive samples. 5 versions of the reduced test set are created varying the negative samples.
[23]	Half million Flickr images weakly labeled with one ANP; Image tweets dataset: Tweets that contain images. The total is 1269 images. AMT is used to generate sentiment labels. Three sub-datasets are created: 581 positive and 301 negative images where 5 labelers agree, 689 positive and 427 negative images where at least 4 labelers agree, and 769 positive and 500 negative images where at least 3 labelers agree.	Randomly chosen 90% of the images from Flickr are the training dataset. The remaining 10% images are used as the testing dataset in the experiments with CNN and PCNN without domain transfer. 5-fold cross-validation is performed with Twitter.com images, using the training images to fine-tune a pre-trained model on Flickr images and the testing images to validate this model.
[24]	Amazon product reviews dataset: contains review comments about a large range of products including books, movies, electronic goods, apparel, etc; Pros and Cons and Comparative Sentence Set databases containing a list of positive and negative sentiment words/phrases; selected 28 youtube videos rated manually (16 negative and 12 positive sentiment) containing expressive speakers sharing their opinion on a wide variety of topics including movies, products, and social issues.	From the combination of the Amazon product reviews, Pros and Cons and Comparative Sentence Set datasets, extracted 800000 reviews for training, and 250000 reviews for evaluation were used.
[25]	EMODB database: Ten professional speakers for ten german sentences, 535 speech files, seven emotions (anger, anxiety, boredom, disgust, happiness, neutral and sadness), recorded at 48 kHz; IEMOCAP database: Audio-visual data in English, only audio track considered for this work, five male speakers and five female speakers, six emotions of the IEMOCAP database are considered (anger, excited, frustration, happiness, neutral and sadness), recorded at 16 kHz; FAU AIBO database: spontaneous emotional speech, contains recordings of 51 German children (21 male and 30 female) at the age of 10-13 years interacting with a pet robot. Contains 9959 training chunks and 8257 testing chunks with length approximately 1.7s. Chunks are categorized into five different emotions (anger, emphatic, neutral, positive, and rest).	Leave-one-speaker-out cross-validation protocol was used for EMODB, IEMOCAP, and FAU AIBO databases. Additionally, with FAU AIBO database, a predefined partition of one children's data is used for validation purposes, and the remaining children's data is used for training purposes.

Table 2.4: Datasets used and partitions for training and testing of the detection approaches (continuation)

Reference	***Dataset or datasets***	***Partitions***
[26]	FAU AEC database: based on the FAU AIBO emotion corpus, which contains recordings of children interacting with a pet robot in German speech. In the training set there are 6601 instances of positive and 3358 negative valence, and in the test set 5792 positive 2465 negative valence; TUM Audio-Visual Interest Corpus (TUM AVIC), Berlin Emotional Speech Database (EMO-DB), eNTERFACE, Speech Under Simulated and Actual Stress (SUSAS), and the "Vera am Mittag" (VAM) database. The age, language, kind of speech, emotion type, number of positive and negative utterances and sampling rate are: children, German, variable, natural, 5823, 12393 and 16 kHz for FAU AIBO; adults, English, variable, natural, 553, 2449 and 44 kHz for TUM AVIC; adults, German, fixed, acted, 352, 142 and 16 kHz for EMO-DB; adults, English, fixed, induced, 855, 422 and 16 kHz for eNTERFACE; adults, English, fixed, natural, 1616, 1977 and 8 kHz for SUSAS; adults, German, variable, natural, 876, 71 and 16 kHz for VAM.	FAU AEC is chosen as target set and the rest are used as source sets.
[27]	Corpus of 3,268 textual review summaries produced by 384 annotators, resulting in 1055 rated as negative, 1600 as positive, and 613 as mixed; CitySearch dataset: 87000 reviews describing more than 6000 restaurant businesses from the citysearch.com website; AMT text dataset: short text reviews summarized by Amazon turkers; GoodRec dataset: a set of short restaurant and bar recommendations mined from the goodrec.com website; short reviews of restaurants dataset: 84 participants made short reviews of restaurants on phone, answering questions, rating them and making a short free review. Resulted in 52 positive and 32 negative reviews.	The text-based classification is done training with the complete set of textual review summaries. The speech recognition models were trained using the CitySearch, AMT, and GoodRec datasets. Sentiment analysis from acoustic features models were trained on the short reviews of restaurants dataset, performing 10-fold cross-validation.
[28]	VAM corpus: consists of audio-visual spontaneous speech. signals were sampled at 16 kHz and 16 bit resolution. Facial image sequences were taken at a rate of 25 fps. Labeled with emotion by human listeners. The signals were sampled at 16 kHz	The VAM corpus was used for all the experiments. 245 utterances of 20 speakers for acoustic emotion estimation were used, performing 10-fold cross-validation. For the visual emotion estimation, 1600 images were used, again performing 10-fold cross-validation. For audio-visual fusion emotion estimation, 234 sentences and 1600 images were used.
[29]	Database for Emotion Analysis using Physiological Signals (DEAP): physiological signals Electroencephalogram (EEG) and Peripheral physiological signals (PERI) are used. EEG was recorded from 32 active electrodes (32 channels). PERI (8 channels) from Peripheral Nervous System (PNS) include Galvanic Skin Response (GSR), Skin Temperature (TMP), Blood Volume Pulse (BVP), Respiration (RSP), Electromyogram (EMG) collected from zygomaticus major and trapezius muscles, and horizontal and vertical Electrooculograms (hEOG and vEOG). The signals were recorded while playing 41 different music clips, and self-report of valence and arousal was done by the participants. Ten participants did 400 self-reports on valence and 400 on arousal.	For the feature-level fusion method, the DEAP database was used in the training and the most significant feature sets were selected for testing. Nested five-fold cross-validation was used in the testing phase. For decision-level, features were extracted in the training as well by performing nested five-fold cross-validation. DEAP database was used for all the experiments.
[30]	Sensitive Artificial Listener Database: spontaneous audio-visual interaction between a human and an operator with different personalities (happy, gloomy, angry, and pragmatic). The Sampling rate for video is 25 fps, and for audio 16 kHz. A set of coders annotated the recordings in the continuous valence-arousal 2D space confined to [-1,1], although not all the data in the database has been labeled.	For validation, a subset of the SAL-DB that consists of 134 audiovisual segments (a total of 30,042 video frames) obtained by automatic segmentation was used. In this work subject-dependent leave-one-out-validation evaluation was used for the experiments.
[31]	eNTERFACE database: 42 non-professional subjects, 81% were male and 19% female, reacting to 5 sentences for each emotion between anger, disgust, fear, happiness, sadness, and surprise. The average length of the samples was 3 seconds. Berlin's emotional speech database and Kanade-Cohn emotional face database were used as single modality databases. Subjects were well-trained for acting according to the emotions. The extra emotion category is found in the Berlin database. Additionally, a massive amount of continuous video (voice or speech and facial video) generated from a video camera or smart mobile-phone based cameras, while the person is using social network service or smart health monitoring services was compiled into five datasets of different sizes.	In the experiments without using big data tools, four-fold validation was performed on the eNTERFACE, Berlin, and Kanade-Cohn databases. In the experiment with big data tools, the five datasets of continuous video generated were used, and a block replication number of three and a block size of 64 MB were used as the default settings in Hadoop. The authors varied the settings in the experiments in order to examine how the performance varies with respect to various: cluster sizes, block sizes, and block replication numbers.
[32]	Kindle Fire HD 7 reviews. Unstructured review data collected from October 2, 2012 to November 20, 2013. User-provided ratings.	A ten-fold cross-validation method is adopted for fine-tuning the parameters of the fuzzy Support Vector Machine (SVM) models and for sentiment prediction, using the Kindle Fire reviews data.
[33]	6 Text user review datasets: IMDB dataset of film reviews (2000 reviews); hotel reviews (2874 reviews); Amazon apparel products reviews (2072 reviews); Amazon music products reviews (2034 reviews); Amazon book products reviews (566 reviews); Amazon electronic products reviews (5902 reviews). All of the datasets have an equal number of positive and negative labeled reviews.	6 distinct case bases are created by training on datasets of all but one of the domains, and then each case base is used to classify documents on the hold out domain, which is the domain not used for populating the case base of the Case-Based Reasoning (CBR) module.
[34]	1999 reviews about digital cameras labeled by users of Amazon with sentiment polarity. 1000 positive reviews and 999 negative reviews; 1991 reviews about DVD movies labeled by users of Amazon with sentiment polarity. 996 positive reviews and 995 negative reviews.	Leave-one-out cross-validation on the two datasets (cameras and movies) was performed.
[6]	Development corpus: this corpus is a collection of 3000 stress-related tweets, manually classified by the author for stress and relaxation. These tweets were identified by monitoring a set of stress and relaxation keywords over a week; six corpora of English short text messages extracted from Twitter.com (tweets), and coded by humans with stress and relaxation strengths. They were extracted from Twitter.com monitoring certain keywords in a 1 month period in July 2015. The corpora are: Common short words (608 tweets); Emotion terms (619 tweets); Insults (180 tweets); Opinions (476 tweets); Stress terms (655 tweets); Transport (528 tweets).	For assigning term strengths, identify missing terms, and to refine the sentiment term scores the development corpus was used. The performance of the supervised version of TensiStrength was evaluated using 10-fold cross-validation 30 times on the data of the six English short text datasets, with the average scores across the 30 iterations recorded.
[35]	24 participants, with ages ranged from 18 to 56, being 14 female, 10 male and 22 right-handed. They were asked to type with a keyboard after cognitive and physical stress tasks. Sessions spread over at least 3 days, ranging from 3 to 22 per participant, with a median of 9 days. The data collected was information about event (key op or down), time stamp (10 ms resolution), and key code. After each task, participants self-reported their stress level.	Baseline condition, control condition, and 2 experimental conditions were used. Baseline: 10 samples under no stress. Control: two samples under no stress. Experimental: completed either a cognitively or physically challenging task prior to providing a typing sample. The performance of each machine learning model was evaluated with three-fold cross-validation.

2.4 MAS-based prevention and recommendation systems review

In the literature, we can find several works addressing user guiding or recommendation using the MAS technologies for it, and in this section, we review works in this line. The problem addressed, techniques, and contributions of each work are summarized in table 2.5. Moreover, the automatic detection of sentiment polarities and stress levels by the system could be used to achieve a more satisfactory and safe user experience, by preventing potential risks that could arise from the interaction (e.g. triggering contact risks by publishing information that you do not really want to post because of cognitive distortions, attracting sexual predators). In this section, previous works in the lines of applying user state detection to risk prevention will be reviewed.

For the works reviewed in this section, the last rule of the inclusion/exclusion criteria presented in section 3 does not apply, since in this section works are presented which apply risk prevention, recommendation, and user-guiding approaches using MAS-based technologies, and thus examine different applications of the existing technologies for addressing these problems. Additionally, some works in this section cannot be directly compared (they address different topics such as user protection against cyber-bullying or on-line grooming and group recommendation, which are completely different problems). Therefore, only the techniques used are analyzed in detail, and not a performance comparison like the one performed with detection approaches in section 3.

In [43] agents and a multi-agent system are suggested to work as communicator mediators between users of SNSs and social groups; in [44] an ontology is constructed by monitoring user behavior, and later used in a task of collaborative filtering recommendation, by means of computing inter-ontology similarities; trust and reputation of agents in a MAS architecture are computed in [45] on the basis of certified recommendations (e.g. based on signed or witnessed transactions), to make the system able to determine how much the agents can be trusted as experts; in [46] an XML-based MAS architecture is proposed, which the authors called MAST. It supports business-to-customer (B2C) e-commerce activities, by means of user personalized profiles that are built and updated by weighting the activities performed in B2C pro-

Table 2.5: Problem addressed, techniques, and contributions of the prevention approaches

Reference	***Problem to address***	***Technique***	***Contributions of the proposal***
[43]	Enhancing communication between users in Social Network Sites (SNSs).	Multi-agent System (MAS) and agents working as mediators.	Enhanced user engagement and collaboration in SNSs.
[44]	Collaborative filtering recommendation.	User ontology created by monitoring user behavior, and calculation of inter-ontology similarities. MAS that integrates the previous tasks, representing users as agents with an ontology representing their behavior.	Automatically creating a model of users by creating ontologies monitoring users, and computing similarities between users using such ontologies for recommendation.
[45]	Trust and reputation in MAS.	MAS implementing agents that perform certified recommendations. The certifications are achieved by using signed transactions or witnessed transactions by other agents as certificate.	Certified recommendations and possibility of the MAS to determine how much agents can be trusted as experts.
[46]	Business-to-customer (B2C) e-commerce activities.	XML-based MAS architecture with users personalized profiles. Such profiles are built and updated by weighting activities performed in B2C processes.	Implementation of business-to-customer e-commerce through using user profiles built with information of user actions in previous transactions.
[47]	Privacy-preserving recommendation systems.	Privacy-preserving protocol for information filtering processes that makes use of a MAS architecture and suitable filtering techniques (feature-based approaches and knowledge-based filtering).	The proposed approach provides information filtering while preserving privacy. An application of the proposal supporting users in planning entertainment-related activities is presented.
[48]	Content-based recommendation system, aiming to solve the new user and overspecialization problems.	MAS architecture as a recommendation system. Semantic enhancement of user preference through domain ontology and semantic association discovery in user profile database.	Addresses two existing problems in an existing technique, and experimental results show an improvement in positive feedback rate.
[49]	Group recommendation.	MAS approach based on negotiation techniques. A multilateral monotonic concession protocol is used to combine individual recommendations into a group recommendation.	Implementation of group recommendation using a MAS architecture and a multilateral protocol. Testing the proposed approach in the movies domain, users were found more evenly satisfied in the groups than with ranking aggregation.
[50]	Detecting the social emotion of a group of entities and influencing them.	Social-emotional model computed using an ANN, based on the pleasure, arousal, and dominance (PAD) three-dimensional emotional space. Application of the model in a group of Human-Immersed agents.	Artificial Neural Network (ANN) that computes the social emotion of a group of agents. Experiments show that using the proposed model to predict the emotion of a group of agents and computing the distance to a target emotion 'happiness' for selecting the action for the system to take achieves the distance to the target emotion to diminish after few iterations.
[51]	Cyber-bullying and on-line grooming prevention in SNSs through the use of different techniques.	Sentiment analysis on text by using different text mining modules, adult image detection using Skin Tone Pixels detection and message classification using Natural Language Processing (NLP) algorithms, through keyword search in the text.	Combination of different data analysis techniques including text and image analysis for prevention of user negative behaviors such as bullying and grooming.
[14]	Prevention of negative outcomes in SNSs, negative sentiment, and high-stress levels through decision-level fusion of sentiment and stress analysis on text.	Sentiment, stress, and combined analysis of sentiment and stress using decision-level fusion on text using ANNs. MAS architecture with agents integrating different unimodal analyses, and an agent performing decision-level fusion and feedback generation to users in SNSs.	Combination of different data analysis techniques and a fusion technique with a MAS architecture for prevention of negative outcomes in SNSs. Experiments with data from Twitter.com that show significant differences between the analyzers predicting negative outcomes, and with a real-life SNS.
[10]	Prevention of negative outcomes in SNSs, negative sentiment, and high-stress levels through decision-level fusion of sentiment and stress analysis on text and keystroke dynamics data.	Extension of a MAS architecture that employs ANNs for sentiment and stress analysis on text with new ANNs performing sentiment and stress analysis on keystroke dynamics data. Design of different decision-level fusion methods employing sentiment and stress analysis on text and keystroke dynamics data.	Addition of analyzers performing sentiment and stress analysis on keystroke dynamics data to a MAS with analysis on text data. Experiments performed with data from Twitter.com exploring different decision-level fusion methods, and proposal of a novel rule-based feedback generation agent in the MAS, in accordance with the results of the experiments.
[52]	Learning the sentiment associated with specific keywords from different data sources.	MAS architecture with agents implementing reinforcement learning algorithms, learning the sentiment associated with keywords, with each agent analyzing data from a different source.	Implements sentiment analysis on keywords by applying collective learning from different data sources and reinforcement learning algorithms in a MAS architecture.
[53]	Sentiment analysis on different SNSs using user opinion to construct a collective sentiment as the opinion of a product.	MAS architecture with agents implementing naïve Bayes classification for performing sentiment analysis on user opinions from different SNSs. A final sentiment is calculated using a common blackboard.	Collective sentiment or opinion about a product computed using sentiment analysis on different SNSs with a MAS architecture.
[54]	Design and implementation of an actor-based software library for building distributed data analysis applications.	Prototype library implemented using the ActoDeS software framework for the development of concurrent and distributed systems. The library implemented includes a MAS architecture and different implementations of five agent types, which are acquirer, preprocessor, engine, controller, and master.	Prototype of a library that provides a MAS architecture and agent implementations that wrap the different tasks of a data analysis application.
[55]	Framework for understanding and predicting the emergence of collective emotions.	Framework built as an agent-based model with agents modeled with individual emotion states and communication between agents.	Proposal of a framework that allows to understand and predict collective emotions, based on interactions between agents, who have individual emotional states.
[56]	Product opinion mining from SNSs data using big data analysis.	MAS architecture including a data extraction, analysis, management and manager agents. It is implemented using JADEX, an agent architecture for representing mental states in JADE agents. Agents make use of Hadoop MapReduce for data process and analysis, and HBase for data storage. Influence of the poster, knowledge about the topic, and sentiment analysis are computed on text messages in MapReduce.	Implementation of distributed data analysis using big data tools for opinion mining from SNSs.

cesses. MAS architectures can be useful for several different purposes. The works already presented in this section showed that they can be successfully used for implementing interfaces between users and social content in SNSs, for guiding users; for monitoring the user behavior and later give advice; for addressing trust and reputation of agents in the system, so users can differentiate between which agents they should trust for a certain task; for addressing B2C e-commerce activities and other activities in which the user interacts with a business in a certain way, so these processes can be more guided. All of these applications of MAS technologies are useful for providing a more guided and satisfactory experience for users in a system in different ways. Additionally, since works addressing these different applications of MAS technologies for guiding users have been selected and reviewed, we followed our inclusion/exclusion criteria.

Moreover, MAS architectures have been used for creating recommendation systems. In [47] an agent-based approach was developed for implementing privacy-preserving recommendation systems. The authors provide a privacy-preserving protocol for information filtering processes and make use of suitable filtering techniques, resulting in an approach that preserves privacy in information filtering architectures. An application of the proposed approach which supports users in planning entertainment-related activities is presented; a MAS architecture is proposed in [48] as a content-based recommendation system, aiming to solve the new user and overspecialization problems existing in such systems. Semantic enhancement of the user preference through domain ontology and semantic association discovery in user profile database are performed, to address the new user problem. Experimental results suggested that there is an improvement in positive feedback rate. A multi-agent approach based on negotiation techniques for group recommendation is proposed in [49]. A multilateral monotonic concession protocol is used in this approach to combine individual recommendations into a group recommendation. Applying the proposal to the movies domain, experimental results showed that applying this negotiation protocol, users were found more evenly satisfied in the groups than using traditional ranking aggregation approaches. To sum up, the MAS architecture proved to be useful for diverse aspects of recommendation systems, such as privacy-preserving recommendation, solving problems existing in traditional recommendation systems, and to perform group recommendation. Since these topics of group recommendation using MAS technologies are different between them, and give an insight about the usefulness of this technology for that purpose, the works [47], [48], and [49] were selected because every one addresses one of these topics, thus following our inclusion/exclusion criteria.

In [50] a social-emotional model is computed using an Artificial Neural Network (ANN) for detecting the social emotion in a group of entities. The model is based on the pleasure, arousal, and dominance (PAD) three-dimensional emotional space. The authors show an example where they use the model to infer the emotion of a group of human-immersed agents when they hear a song, then predict the future emotion that the group would achieve after hearing new similar songs, and finally compute the distance of the predicted emotion to the target emotion 'happiness' for deciding which song to play, that would be the one that minimizes this distance. The reported distance of the emotion of agents detected against the target emotion on the experiments performed quickly diminishes after a few iterations of the system. Sentiment analysis was used in [51] on the texts of users interacting inside a SNS, together with adult image detection and message classification to help the system ban users that incurred in either cyber-bullying or on-line grooming. Moreover, in [14] a set of analyzers were built for performing sentiment analysis, stress analysis, and a combined analysis of sentiment and stress, using sentiment and stress on the text messages of users interacting in a SNS. When the system detects negative sentiment or high levels of stress, a warning is sent to the user to prevent him from sending the message to the network as it is and avoid potential negative outcomes in the SNS. The authors performed experiments with data from Twitter.com to discover which analyzer of the proposed ones was able to detect a state of the user that propagated more to its replies in the SNS, finding significant differences between the analyzers. Experiments with a private SNS called Pesedia [12] were also performed to test the system in a real-life scenario. Additionally, in [10] new agents were added to the system presented in [14] to perform sentiment and stress analysis on keystroke dynamics data, proposing fusion analysis that employed analyzers working on text data and the ones working with keystroke data. Experiments were performed with data gathered from the private SNS Pesedia to find which of the proposed analyzers worked best at detecting states that propagated more in the SNS. Finally, a new version of the advisor agent was proposed, which generates feedback to users based on the input of the analyzers found best in the experiments. Thus, the MAS architecture is not only useful for guiding users or recommending them, but it also proved to be able to detect the state of users and use it for helping to prevent issues that could arise in a social environment, or make for a better social experience. This can be seen in [50], where the goal of the system is to detect a group emotion and simulate agents interacting to achieve a better social experience; in [51] where the system detects content in images and text to help itself detect dangerous users; in [14] and [10], where the system analyzes the sentiment and stress levels of users to help prevent negative interactions in SNSs and risks. These four works were selected for assessing the usefulness of MAS technologies for detecting the state of the user to address the different topics mentioned. For the case of [14] and [10], the former addresses sentiment and stress analysis on text data for guiding users, while the latter uses both text and keystroke data combined for this purpose.

A MAS with agents that apply reinforcement learning algorithms to learn the sentiment pertaining to specific keywords was presented in [52]. Agents collectively learn this sentiment since each agent processes its assigned subset of data. Experiments were conducted on abstracts from PubMed related to muscular atrophy, Alzheimer's disease, and diabetes, and results show that the system was able to learn the sentiment score related to specific keywords. A MAS architecture with a set of agents that work with opinion data from different SNSs is proposed in [53]. The proposed system has an agent extracting opinions about a product from Twitter.com, another from Wikipedia, and another from Facebook. All agents compute sentiment using machine learning techniques and are able to communicate between them by using a blackboard. In this way, they can generate a more complete opinion about the product with sentiment computed on additional features by other agents. In [54] ActoDatA (Actor Data Analysis), an actor-based software library for building distributed data analysis applications is presented, and a prototype is implemented. This library provides a multi-agent architecture and different implementations of five agent types, which are acquirer, preprocessor, engine, controller, and master. Each agent acts as a wrapper for components that perform the different tasks of a data analysis application. A framework built as an agent-based model is presented in [55]. The framework is used to understand and predict the emergence of collective emotions based on interactions between agents, who have individual emotional states. For helping enterprises be aware of their customer's opinions about products or services, an agent-based social framework that extracts reviews from social media is presented in [56]. Data analysis and storage are performed by using a framework based on Hadoop MapReduce and HBase, respectively, which allows the efficient manipulation of big amounts of data, which is the case for the task of opinion mining from SNS data. As can be seen, the MAS architecture can be useful to model the emotion of users or mine opinions in different ways that are related to computing this emotion or opinion in a distributed way. It can help compute collectively emotion using different sources of data, can help build distributed data analysis applications, predict the emergence of collective emotions based on interactions between agents with individual emotions, or create big data distributed applications to perform opinion mining with big volumes of data, such as SNS data. Therefore, the emotion of users can be computed in different ways and in a distributed architecture using MAS technologies and machine learning or other detection technologies, and then use this information to guide users, recommend, and prevent risks and potential issues in SNSs or other environments with emotional entities. For addressing the different topics presented related to using MAS technologies to detect emotion or mine opinion in a distributed way, a set of works was selected according to our inclusion/exclusion criteria: [52] was selected for presenting a MAS with agents applying reinforcement learning algorithms to learn the sentiment of associated to keywords from different data; [53] was selected for addressing distributed learning of a product opinion from different SNS data; [54] presents an actor-based library for building distributed data analysis applications; [55] addresses the problem of understanding and predicting the emergence of collective emotions in the basis of interactions between agents; [56] addresses opinion mining from SNS data using big data techniques.

2.5 Discussion

As has been reviewed in the previous section, there is a substantial effort in the literature on detecting the sentiment polarity of people that create different kinds of content, using different sources of data. Between the data used to perform sentiment analysis, we can find extensive literature featuring approaches using text, audio, images, and in less quantity writing patterns. Some approaches only perform detection of sentiment on one data source, while others do it on multiple, which is the case of multi-modal sentiment analysis. CBR technology has also been applied to performing sentiment analysis. Additionally, stress level detection has been performed on text data and writing patterns. Moreover, as has been commented in section 1, the emotional state affects the decision-making process, and there are risks that can arise from social interaction in on-line social environments such as SNSs. User state detection, as has been revised on this survey can be effectively used together with MAS technologies to guide and recommend users, and prevent potential risks or issues. Moreover, using different data modalities, which can be implemented in the MAS architecture, can improve the capacity of the system to detect risks.

In the literature, there are works reviewing different previous approaches for sentiment analysis. On the one hand, there are surveys specialized in single data modality. Aspect-based sentiment analysis works on text data have been reviewed in [37], while works on sentiment analysis applied to images have been reviewed in [38]. On the other hand, [9] is a survey on works about sentiment analysis applied to different data modalities and also multimodal sentiment analysis. Although there are works that review sentiment analysis using one or various data modalities, to the best of our knowledge, there is a lack of a review of works in user risk prevention when navigating on-line social environments, reviewing different strategies of sentiment and stress analysis, which is the case of the present survey. We also focus on works that use MAS-based techniques for recommending and guiding users, which is a technology that can be used together with SNSs as it fits the social network architecture by using agents to represent entities in the network, and guide or interact with users.

2.6 Conclusion and future lines of work

Since we have discussed previous works in risk prevention, recommendation, and user state detection, and highlighted the relations between them and potential uses for preventing users from suffering negative consequences from their interaction, and for helping improve future systems, following potential future works in three different lines will be discussed, which are:

- Using current technologies in user state detection for creating improved user-guiding systems in on-line environments.
- Combining different technologies compatible with the architecture of SNSs with emotion detection techniques and testing their effectiveness in real-life scenarios as guiding or recommendation systems.

- Improving user state detection techniques.

Regarding future lines of work using current technologies in user state detection, it should be taken into account that automatic user state detection gives information to the system about a factor that directly influences decision making, and consequently the probability of incurring one of the risks that arise from the interaction between users. Therefore, it is desirable that a system that guides users navigating exploits the potential of the extensive amount of sentiment analysis techniques in the existing literature. Secondly, combining different sources of data has been shown to help improve the system performance when detecting emotional states, and also using sentiment analysis together with stress analysis. In this way, future works could investigate new ways of combining different kinds of data in fusion approaches, and also use different analysis such as stress analysis together with sentiment analysis to try to improve the system performance. It could allow researchers to discover a correlation between different variables such as stress levels of users, sentiment, or other factors, and at the same time it might improve the system performance to use detection of multiple aspects of the user state to perform a guiding or recommendation, therefore it is an interesting possibility to test.

Another potential branch of future lines of work where there is room for improvement is investigating the effect of a combination of different technologies compatible with the architecture of SNSs with emotion detection techniques, and the effectiveness of such systems in different real-life scenarios (to guide users by analyzing their data and giving them feedback). CBR techniques, as has been reviewed, can be used to work together with sentiment analysis techniques for example exploiting different sentiment lexicons. Since CBR systems can be integrated easily into a SNS as a guiding module that helps users by monitoring their interactions and giving them feedback based on the different characteristics of the interaction and user states, they are potential candidates for designing user-guiding systems that create a more satisfactory and safe user experience. Moreover, recommendation systems that use system data to give recommendations to users could potentially be improved, using, for example, persuasion techniques and sentiment or stress analysis. Finally, MAS-based approaches can fit in the SNS architecture, for example assigning agents to users and different system tasks to other agents. These systems can work together with user state detection techniques as user-guiding and recommendation systems, by using the data collected by the automatic user state detection when users interact to give advice or recommendations. Therefore investigating the effect of combining different user state detection techniques with MAS-based approaches might improve the performance of the system as a guiding or recommendation system. It might be interesting to also test other technologies such as Peer-to-peer and Internet of things to work as guiding and recommendation systems, integrated into a SNS or other social environment, since they can allow users to share information in a distributed way, and the system could use this information to perform its guiding and recommendation functions.

Finally, related to the user state detection techniques, new fusion techniques, that use feature-level, decision-level, or hybrid fusion could be tested to analyze a potential improvement in emotion detection, as has been shown in the literature that some approaches manage to beat the accuracy of non-fusion techniques with a fusion technique using the same data. It might be an interesting research to also apply fusion techniques to stress analysis and other aspects of the user state (e.g. fusion of text, keystroke dynamics data, and images to determine the tiredness of users, or to perform fusion of image data and text data to determine the level of interaction with other users that a given user has in a SNS), and see if there are differences in accuracy achieved between unimodal and multi-modal techniques. To summarize, the recent literature contains several works in both automatic user state detection and user guiding and recommendation in on-line social environments, but there is still plenty of potential improvement in those lines of work, by improving the accuracy of user state detection models and testing the usefulness of different technologies and combinations in user-guiding and recommendation systems. Moreover, improving the feedback given to users to create a better understanding of potential risks and a better response of the users for avoiding them could be implemented in guiding systems.

CHAPTER 3

MAS for sentiment, stress, and combined analysis on text data

3.1 Introduction

Multi-Agent Systems (MAS) are systems composed of multiple software agents that interact together, and they are usually used to address computational problems that require different software entities to be independent and interactive such as on-line trading or complex simulations with multiple entities, like social response modeling. In this chapter, the goal of the proposal is to create a MAS that will be integrated into a social environment, like a social network, and incorporates different analyzers, and a combined version of analyzer. Those analyzers will be able to recognize the emotional state of the users in the social environment, their stress level, and to advise them to post or not by analyzing text messages before they post them. Finally, we performed experiments with data extracted from Twitter.com, to be able to discover which analyzer is more suitable for predicting a future negative outcome in the social environment (potential negative outcome caused by negative states detected in the text messages), and in which cases. In the experimentation, we use the replies of text messages to discover if the emotional state or stress level detected has propagated or not from the original messages to the direct follow-up replies, and to study which analyzer is able to detect a state that propagates more in the network. We designed a MAS which includes a set of agents in charge of performing different kinds of analyses (sentiment, stress, and combined), and that interacts with the users advising them at the moment of publishing a message. This system is integrated into a Social Network Site (SNS) to perform advice to the users according to their detected state, in order to help them with their decision-making process, and for avoiding potential future negative situations that could arise as a result of their interaction in on-line social environments.

The rest of the chapter is structured as follows. Section 2 presents an introduction to recent work in the lines that are related to the chapter work, such as sentiment and stress level detection and user state modeling. Section 3 describes our proposed MAS and its agents, which perform sentiment analysis, stress analysis, and combined analysis, and the advisor agent that performs warnings to the users using the information from the analyses. Section 4 exposes experiments conducted with corpora of text messages from Twitter.com, which aims to calculate which analyzer is able to better predict the repercussion that messages with a label of negative sentiment or high level of stress have in the network, and in which cases. Section 5 shows the results of the experiments presented in section 4. Finally, section 6 presents conclusions and the next iteration of the system proposed.

3.2 Related Work

The present chapter deals mainly with two research areas, which are sentiment analysis and stress analysis, but it is also related to user state modeling and MAS. Following, we analyze recent works in those areas.

Sentiment analysis is a field of research that intends to study the phenomena of opinions, sentiments, evaluations, appraisal, attitude, and emotion through different kinds of media (e.g. written messages, images, emoticons, etc.) [57]. Regarding sentiment analysis in written texts (which is the most common), we will find that there are four well-differentiated techniques: document-level sentiment analysis, sentence-level sentiment analysis, aspect-based sentiment analysis, and comparative sentiment analysis [36]. The kind of analysis depends on the level of fine-grained sentiment analysis that we choose to perform, starting out from the document-level sentiment analysis (sentiment from the entire document), to sentence level (sentiment in a sentence), and finally to the aspect-based sentiment analysis (sentiment in concrete aspects, as sequences of words that can be one word, found in the text). Comparative sentiment analysis is an exception, where we use comparative sentences to learn which are the preferred entities, associated with comparative words appearing in the sentences (the sentiment words for the model) [36]. For the present study, we choose to use aspect-based sentiment analysis on texts so we can perform a fine-grained analysis, focusing on terms that may contain sentiment and not entire sentences or documents, which may contain more than one.

Two different work lines are present in sentiment analysis, that sometimes are worked in hybrid approaches, and those are aspect extraction and sentiment classification [37]. For aspect detection, we can find detection through generative models (e.g. Conditional Random Fields or CRF), that use a variated set of features [17]; frequency-based methods, which use the frequency of the terms in the training corpus to put them as aspects or not in the aspect set (the most frequent terms are added) [16]; non-supervised machine learning techniques (e.g. Latent Dirichlet Allocation or LDA) [58].

In this chapter, we use a frequency-based method because it helps to know what aspects are the most frequently mentioned in an SNS.

In the case of sentiment classification, there are different methods like machine learning methods, which can be either supervised or not, and dictionary-based methods. Machine learning methods use Support Vector Regression and other techniques to obtain the features for training the model, and non-supervised methods use other techniques like relaxation labeling [37]. Dictionary-based methods use a dictionary of aspects with a polarity assigned to them in the training step (with a method for training the aspect set), and a method for extracting polarities later from texts using the dictionary [37]. We choose a dictionary-based method because that way we will be able to have a set with sentiment aspects and another with stress aspects (sequences of words), with associated sentiment polarities or stress levels. Finally, regarding the hybrid approaches, they intend to detect aspects and assign polarities at the same time [37], but those are not used in the work presented in this chapter since we want to simplify the process of detecting aspects and assigning polarities and to modularize it.

TensiStrenght [6] is an algorithm derived from the SentiStrenght [7] algorithm for sentiment strength detection, that uses a set of terms associated with stress and another set of terms associated with relaxation. Those are previously trained assigning levels of stress and relaxation to its aspect sets with an unsupervised method that uses tweets annotated with stress and relaxation strengths, and then refining the values with a hill-climbing method. The sets are then used to detect stress and relaxation levels in sentences of written texts, with some improvements implemented in the algorithm such as detecting exclamation marks and boosting the strength of stress or relaxation within a sentence.

In the case of works trying to model the information of the user on a system, Rincon et al. [50] created a social-emotional model that detects the social emotion of a group of entities. They used the Pleasure, Arousal, and Dominance (PAD) three-dimensional emotional space for representing the emotions of the entities and an Artificial Neural Network (ANN) to learn the emotion of the group in the context of an event that just happened; Gao et al. [59] used a model for a task of sentiment classification that computes the user and product-specific sentiment inclinations; in [60] a nearest-neighbor collaborative approach was used to train user-specific classifiers, which were finally combined with user similarity measurement in a sentiment analysis task.

MAS for helping or guiding users have been worked on before. A MAS-based system named PATRASH (Personalized Autonomous TRAansit recommendation System considering user context and History) was presented in [61]. It was designed as an application for public transit guide system, where each agent interacts with each other by exchanging messages of CSV-format.

The contributions of the present chapter are a design of agentized versions of a sentiment analyzer and a stress analyzer; the design of a novel agentized combined analyzer; experiments that aim to discover what agent performs better at predicting a potential future bad outcome, and in which cases, by calculating the value of the different analyses on short text messages; the design of a MAS that integrates the analyzers and uses those values to advise the users of an on-line social environment such as a SNS. With this work, we will be able to help build intelligent systems integrated into on-line social environments like SNSs, where there are several users interacting together. Such systems will be able to help them in their experience, by guiding their decision-making processes (e.g. warning them when they could generate a problem by means of the different analyses of data).

3.3 Description of the proposed MAS

The system has been designed as a MAS that will analyze data from written text messages so it can give recommendations and warnings to the user for helping in their social experience. We designed the system as agent types, that are components of the MAS. These agents perform different tasks on the system and communicate with other agents in order to accomplish their tasks. They use the SPADE 2.3 multi-agent platform [62], which is distributed under a GNU Lesser General Public License v2 (LGPLv2), for their implementation.

The MAS proposed has three layers, which follow a presentation, logic, and persistence layers architecture. The system is structured into diverse agent types that operate in different layers and each one has a different task to perform. The presentation layer has an agent type to show information to the user and to get the information of the user and send it to the logic layer. The logic layer has agent types that perform the analysis and calculations of the system and generate recommendations or advice for the users. These agents get input from the presentation and persistence layers and send information to the persistence layer for storing it. Finally, the persistence layer has the agent type that stores the data into the database and provides it to the logic layer when it is needed. The architecture of the MAS can be seen in figure 3.1. As it can be seen, the advisor agent can either get the information of sentiment polarities and stress levels from the analyzer agents, when a user is posting on the SNS or from the database.

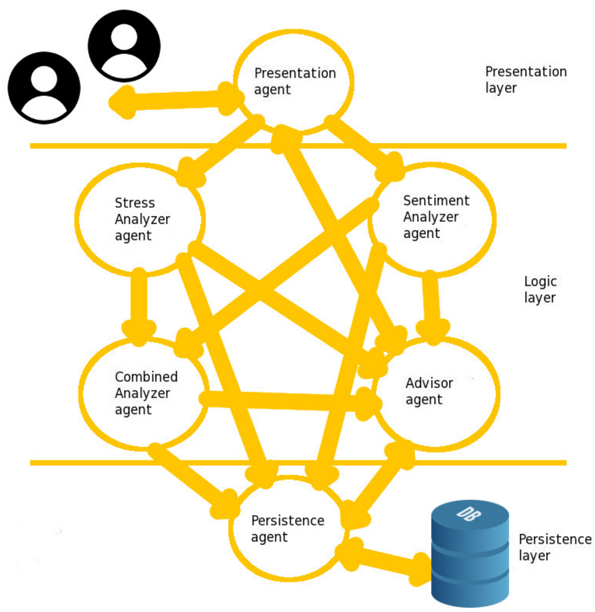


Figure 3.1: Architecture of the system built as a MAS with three layers

As we can see in figure 3.1, there are three different analyzer agents, that correspond to the sentiment analyzer agent, the stress analyzer agent, and the combined analyzer agent. Each of those agents will be performing a different kind of analysis on the system. The combined analyzer agent will be interacting with the other two agents to perform a combined analysis. Other auxiliary artifacts have been built, to pre-process and translate tweets and to extract data from Twitter.com We will explain in more detail the agent types of the MAS in the next subsections.

3.3.1 Presentation agent

The presentation agent type uses different widgets for getting information from the users on the SNS, so it can get the text messages or other data, and send it to the analyzer agents in the logic layer. It also can get information about advice or warnings from the advisor agent and show a warning, telling the user to be careful.

3.3.2 Sentiment analyzer agent

The sentiment analyzer agent type is the one in charge of taking the short text messages as input (e.g. from a social network), analyzing it using a previously trained aspect set, and giving, as a result, its sentiment polarity (positive, negative, or neutral). This agent sends this polarity to other agents for storing it in the database or for using it in more calculations and generate advice or warnings for the users. For designing it, various decisions have been made:

- Aspect-based sentiment analysis: The kind of sentiment analysis chosen for the system was aspect-based. This type of sentiment analysis, as explained in the previous section, performs an analysis based on concrete aspects found in the sentences of texts, creating the model as an aspect set with associated polarities and later using it to perform the classification of text messages. We used an annotated dataset with polarities assigned to short written messages (tweets), extracted from diverse variated topics (e.g. politics) for training the model. This dataset is extracted from the TASS experimental evaluation workshop [63, 64].

- Aspect extraction: We selected a frequency-based method for performing the aspect extraction, where we create aspects as the terms found in the training corpus, which are unigrams. We select then the terms or aspects with a higher frequency of appearance in the corpus to constitute the aspect set.

- Sentiment classification: Since we have an annotated corpus of data with sentences labeled with a polarity, we classified the aspects of the aspect set using those labels, assigning to them a polarity as the one with a major appearance on the training labeled corpus (the corpus assigns polarities to sentences, so we took those polarities as associated with the terms appearing in the sentence), which means that we use a Bayesian classifier.

- Sentence classification: For using the model we perform a classification of short written texts as follows: All the possible n-grams of the message are compared with each aspect of the aspect set, and if an aspect is found we store that information. Finally, when all the aspects of the aspect set are compared, we determine the sentiment of the message as the most predominant polarity found from the previous exploration. Either positive, negative, or neutral.

3.3.3 Stress analyzer agent

The stress analyzer agent type is similar to the sentiment analyzer agent type, but it assigns levels of stress to the aspects of the aspect set instead of sentiment polarities. In that manner, we can find low-stress level, normal-stress level, and high-stress level associated with an aspect. The dataset used to train the model is also a dataset of messages written in a context in which stress is normally present. This dataset was composed of stress-related tweets annotated with stress strengths, coming from the work on TensiStrength [6], extracted from Twitter.com monitoring a set of stress and relaxation keywords.

3.3.4 Combination analyzer agent

For designing this agent, we use the combined values of sentiment and stress from the text messages. We determined that when stress is in low or normal levels, we assign the polarity of the message as the polarity of the sentiment analysis, but when the stress levels are high we directly assign the polarity of the message being analyzed with this combined model as negative. This is done in this way to determine the effect of high levels of stress in the repercussion of a message in an SNS, we don't experiment with normal levels of stress as considering them negative because those are present in a multitude of situations for different reasons and we choose not to consider a medium level of stress as a negative state. Instead, we study the negative bad repercussions and user negative states as high levels of stress or negative sentiment polarity.

3.3.5 Advisor agent

The advisor agent type performs a calculation based on the information returned by the analyzer agents, calculated at the moment when a user is posting a message in the system. It has been designed to work as follows: first, when the user posts a message, it is sent to the logic layer by the presentation agent, so it is sent to every analyzer. The analyzers perform their analysis on the data and send the result to the database and to the advisor agent. Finally, this agent, taking into consideration the output of every analyzer and the current domain of application (stress present environment or not), will decide whether to send a warning to the user or not if it determines that the message may be showing a potential negative repercussion in the on-line social environment. For deciding what to take into account, regarding the output of the analyzers, we aim to discover what agent works best in each situation, so we conducted experiments that have this purpose, which will be shown in the next section.

3.3.6 Persistence agent

The persistence agent type has functions for storing polarities, text messages, and other data in the database. This agent has also functions to extract information from the database when an agent from the logic layer requests it and can either get or send information from or to the agents on the logic layer.

3.4 Experiments with data from Twitter.com

In this section, we will explain the experimentation conducted and the corpora of data that we used for performing it. We will also show the metrics used.

3.4.1 Design of the experiment

We have taken corpora of data extracted from the popular SNS Twitter.com, composed of text messages from real people from all around the world and characterized for having a thematic on each corpus (e.g. political, cultural). The corpora have been created using the Twitter.com API for streaming tweets, and they have been processed using a function to clean them up for the sentiment, stress, and combined analyzers, which searches for possible sources of error and corrects them. Two corpora were used for the main experimentation:

- Podemos (A political corpus made of messages related to the political party 'Podemos'). This is a very large corpus (about 1.9 million tweets).
- Star Wars (A leisure corpus about the famous franchise of films Star Wars, which gathers tweets from people from all around the world). It is a very large corpus (about 12 million tweets), but only a small part of it is messages in Spanish, and we need them to be in that language because the aspect sets of the analyzers are in Spanish only.

We used also the annotated corpus Stompol [63, 64] for the calculation of the recall of the analyzers. STOMPOL (corpus of Spanish Tweets for Opinion Mining at aspect level about POLitics) is a corpus of Spanish tweets prepared for the research in the task of opinion mining at aspect level.

In this work, we will try to determine the effect that the messages detected as negative or dangerous by the hand of the sentiment and stress analyzers have in the messages that are a repercussion of them (we used the replies of the messages in this case).

We also want to determine the effect that tweets analyzed by our combined model (which uses different analyzers), have on the replies as well, and compare this effect to the effect that we observed in the previous stage (using only one analyzer at a time). With this information, we aim to determine whether it is more useful or informative to use only one analyzer or both combined, and in what situations.

We coded a function that reads and loads a tweet in JSON format and then analyzes it for knowing if it is a Spanish tweet (since as mentioned before, our aspect sets of the analyzers are in Spanish only), and if it is a reply of another tweet. If it passes the two filters, then we proceed to look at a dictionary where we store the analyzed replies, and if the original tweet that generated a reply is not present, then we search that tweet using the

Twitter.com API. We calculate the sentiment and stress of both messages and store those values in the dictionary (if the original message was already present we only calculate the ones of the reply).

When we have all the corpora analyzed, for all the tweets that generated replies we do as explained following:

- Calculate its combined value using both the sentiment and stress value in the way we explained in subsection 3.4.
- Calculate the most present value of sentiment polarity in the replies of that tweet.
- Calculate the most present value of stress level found in the replies of that tweet.
- Calculate the combined value of the replies using both most present values previously calculated.

With this done, we finally proceed to calculate which tweets correspond to its replies in terms of comparing their calculated values (using sentiment, stress, and combined values), with the calculated values in the replies (using the most present value of sentiment, stress level, and combined value on the replies). If it is the same value (positive, negative, or neutral for sentiment, stress levels, or combined value), we conclude that this tweet has generated a repercussion, according to what the model predicted.

Finally, we accumulate the percentage of tweets that are in line (have the same detected emotional polarity, stress level, or combined value), with the output of the analyzers for their replies (using again sentiment, stress, and combined values), and store it as the result of the experiment.

3.4.2 Metrics of the experimentation

We performed an experiment with an annotated corpus with tweets associated with a sentiment polarity, in order to discover the recall of the analyzers. For calculating the recall we took the tweets that were classified as negative by the classifier and the total amount of tweets annotated as negative (or annotated as having a stress level associated as negative by the classifier). We used the following metrics for calculating the result of the experiments:

- For sentiment analysis, percentage of concordance sentiment (PCsen):

  tweetsConc = Amount of tweets with the same emotional polarity than the most present in its replies.

  tweetsTotal = Amount of total tweets with replies analyzed.

  $$PCsen = \frac{tweetsConc}{tweetsTotal}$$

- For stress analysis, percentage of concordance stress (PCstr):

  tweetsConc = Amount of tweets with the same stress levels than the most present in its replies.

  tweetsTotal = Amount of total tweets with replies analyzed.

  $$PCstr = \frac{tweetsConc}{tweetsTotal}$$

- For the combined analysis, percentage of concordance combined (PCcomb):

  tweetsConc = Tweet messages with the same combined value (as the output of the combined analysis) as the predominant combined value calculated in their replies.

  tweetsTotal = Amount of total tweets with replies analyzed.

  $$PCcomb = \frac{tweetsConc}{tweetsTotal}$$

- Recall for the sentiment analyzer (RecallSA):

  NegativeTweetsDetected = amount of tweets considered negative that the analyzer detected.

NegativeTweets = Amount of tweets considered negative in the corpus.

$$RecallSA = \frac{NegativeTweetsDetected}{NegativeTweets}$$

- Recall for the stress analyzer (RecallStr):

  NegativeTweetsDetected = amount of tweets considered negative that the analyzer detected (which in this case is associated with the stress level considered negative).

  NegativeTweets = Amount of tweets considered negative in the corpus (again it is associated with the stress level considered negative).

$$RecallStr = \frac{NegativeTweetsDetected}{NegativeTweets}$$

- Recall for the combined analyzer (RecallCombined):

  NegativeTweetsDetected = amount of tweets considered negative that the analyzer detected.

  NegativeTweets = Amount of tweets considered negative in the corpus.

$$RecallCombined = \frac{NegativeTweetsDetected}{NegativeTweets}$$

3.4.3 Plan of the experiments

We will explain in this subsection how many experiments we launched, of what kind, and with what corpus of data. As stated above, we launched an experiment with a corpus of annotated tweets called Stompol for calculating the recall of the analyzers. We used the number of tweets classified as negative and that actually had a negative polarity label coded by a human, and the total amount of tweets coded as negative. The fraction of negative tweets detected by the analyzer (NegativeTweetsDetected), and negative tweets in the corpus (NegativeTweets), are shown following for the three analyzers:

$$RecallSA = 79.12\%; RecallStr = 0.012\%; RecallCombined = 79.12\%$$

The stress analyzer agent has a very low recall, this may be caused because we only use the high levels of stress to determine whether a user has a dangerous stress level or not. Nevertheless, it has proved to make a difference in the tests over a large amount of data (which will be shown following). The low recall of this analyzer makes it less useful to be used alone, because even if it has a good concordance with the replies (PCstr), it may only detect a small number of negative stress states. Thus trying to combine it with the sentiment analyzer may make it more useful.

The recall for the sentiment analyzer resulted to be the same as the RecallCombined, this may be caused because the Stompol corpus, which was used for the calculation of the recall of the analyzers is a small corpus, and the small number of detections from the stress analyzer was also detected (mostly or completely) be the sentiment analyzer.

In order to get rid of any ambiguity before proceeding, we remind the reader that the analyzers and thus, the system, predict the propagation of a sentiment or stress level in general (e.g. if the sentiment in the original tweet is positive, will this propagate to the replies? and if negative, will it?), it does so in the experiments as well, where what we compare is the sentiment or stress level of the replies to the one in the original message, to know if this sentiment or stress level has propagated to the direct follow-up replies.

Then, the probability of a negative sentiment arising from positive and vice-versa, and all the combinations with different sentiment or stress levels, are just the inverse probability of the one calculated in the experiments, which is the probability of the propagation of the sentiment or stress level from the original to the replies (Amount of times that the sentiment or stress level propagated / number of total comparisons).

- Experimentation with the corpus Podemos. We prepared experiments with the corpus Podemos in the following way: we partitioned this corpus and, since it is a very large corpus, we decided to make six different partition sizes, doing four different experiments for each partition size. This was done in this way because the largest partition size was 1/4 of the corpus replies, and the maximum amount of parts that we could perform without using a tweet more than one time was four. We performed each experiment using the three different analyzers. The first partition is 1/128 of the total replies of the corpus for each experiment (around 1700 replies); the second partition is 1/64 of the replies; in this same way, the following four partitions are of 1/32, 1/16, 1/8, and 1/4 of the total replies, and the final results of the experimentation can be seen at table 3.1.

- Experimentation with the corpus Star Wars. We prepared the experimentation for the Star Wars corpus as shown following: we made partitions of the corpus with four different partition sizes and with three different experiments for each one, and we performed the partitions in this way because even when the corpus is large, the amount of tweets in Spanish is not high, resulting in a modest amount of replies in Spanish (22543 replies). Remember that since the aspect sets of the analyzers are built with aspects in Spanish, we can only analyze Spanish tweets with them. Again, the maximum amount of different experiments that could be performed with the biggest partition size, without using data in more than one different experiment was used for all the partition sizes (in this case 3). For each experiment, we used the three analyzers, and the number of replies for each partition size was 1/3, 1/6, 1/12, and 1/24 of the total replies in the corpus. The final results of this experimentation are shown in table 3.2.

We show the results of the experiments launched for the corpus Podemos in figure 3.2 and 3.3, and the values of the experiments for each corpus size have been represented as one single point as the average of all the experiments launched for that corpus size. We separated the information about the stress analyzer experiments from the others because the percentage of concordance of this analyzer (PCstr) is very high and made it difficult to appreciate well the results of the others when they were shown in the same figure. Finally, we show the results for the experimentation with the corpus Star Wars in figure 3.4 and figure 3.5, in the same way that we did in the case of the experimentation of the corpus Podemos. In the following two figures, the legend stands for:

- SA and Stress A: Sentiment analysis combined with stress analysis.

Table 3.1: Experimentation with the corpus Podemos

Partition size	***Experiment***	***PCsen***	***PCstr***	***PCcomb***
1/128 of replies	1	0.5975	0.9752	0.5944
	2	0.5594	0.9752	0.5644
	3	0.5881	0.9611	0.5943
	4	1.0	1.0	1.0
1/64 of replies	1	0.5789	1.0	0.5789
	2	0.4583	1.0	0.4583
	3	0.5680	0.9813	0.5697
	4	0.4706	1.0	0.4706
1/32 of replies	1	0.5	0.9833	0.5
	2	0.5682	1.0	0.5682
	3	0.5261	0.9799	0.5281
	4	0.5824	0.9780	0.5824
1/16 of replies	1	0.5132	0.9737	0.5
	2	0.5156	0.9778	0.52
	3	0.5616	0.9726	0.5616
	4	0.5375	0.95	0.525
1/8 of replies	1	0.5508	0.9786	0.5508
	2	0.5546	0.9738	0.5611
	3	0.5493	0.983	0.5511
	4	0.5864	0.978	0.5864
1/4 of replies	1	0.5591	0.9694	0.5577
	2	0.5948	0.9752	0.6020
	3	0.5638	0.9741	0.5618
	4	0.5674	0.9787	0.5686

- SA: Only sentiment analysis.
- Stress A: Only stress analysis.

Table 3.2: Experimentation with the corpus Star Wars

Partition size	***Experiment***	***PCsen***	***PCstr***	***PCcomb***
1/3 of replies	1	0.6107	0.9905	0.6069
	2	0.6	1.0	0.6
	3	0.6373	0.9707	0.6327
1/6 of replies	1	0.6075	0.9791	0.6045
	2	0.6209	0.9783	0.6137
	3	0.6275	0.9902	0.6373
1/12 of replies	1	0.6075	0.9794	0.6075
	2	0.6391	0.9699	0.6165
	3	0.6142	0.9864	0.6142
1/24 of replies	1	0.6061	0.9865	0.6044
	2	0.52	0.98	0.5133
	3	0.625	0.875	0.625

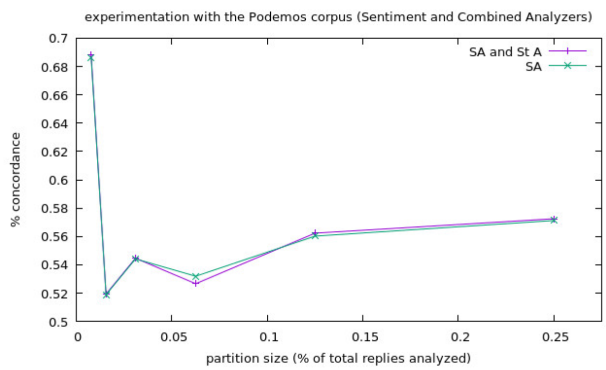


Figure 3.2: Results with the corpus Podemos for the sentiment analyzer and the combined analyzer

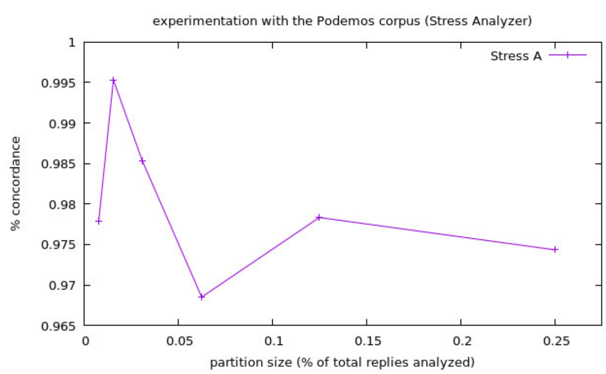


Figure 3.3: Results with the corpus Podemos for the stress analyzer

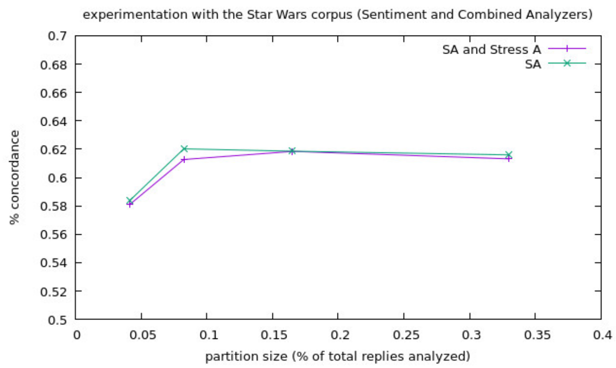


Figure 3.4: Results with the corpus Star Wars for the sentiment analyzer and the combined analyzer

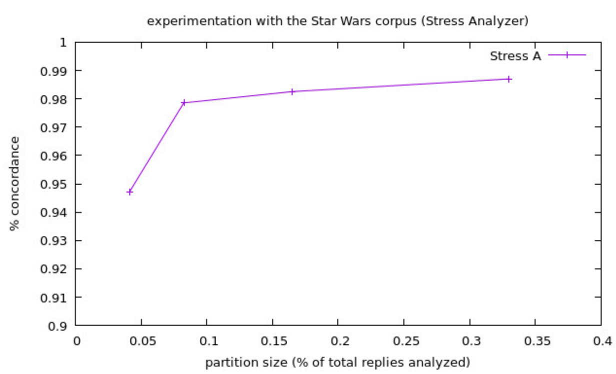


Figure 3.5: Results with the corpus Star Wars for the stress analyzer

3.5 Results

In this section, we will discuss the results of the experimentation with the data from Twitter.com and the different analyzers. We discovered that all the analyzers separately (sentiment, stress, and combined) are successfully able to predict a bad outcome through bad emotional states, high levels of stress, or negative combined value in the text messages. This can be seen in the experiments with all the corpora. Regarding the stress analyzer, despite having a general tendency of high concordance with the replies (PCstr), we have to remember that it has a very small recall (RecallStr) so it may be less suitable than the other analyzers in a variety of cases.

In the case of the Podemos corpus, we can see that there is a big variation at the smallest sized experiments (1/128), but the results are considerably more stable at the big partition size experiments, starting to get more variation again when it comes to the biggest size experiment. This could be caused by the excessive amount of information when more and more replies are added to the analysis. As we can see in figure 3.2, there is a general tendency (except for one case, which is the 1/16 partition size), of the combined analyzer (PCcomb) to perform better than the sentiment analyzer alone (PCsen), from what we can conclude that at least at the domains where there is stress involved (such as politics in this case), the combined analyzer performs better than just the sentiment analyzer. We can see a general tendency of the stress analyzer to fluctuate in the 90% to 100% range of concordance (PCstr).

Regarding the case of the experiments with the Star Wars corpus, we can see a big change from 58% to 61% approximately for the combined analyzer, from 58% to 62% for the sentiment analyzer, and from 94.7% to 97.8% for the stress analyzer, between the experiments with the smallest partition size (1/24), and the second smallest one (1/12), but stable results in the rest of the experiments. This is in line with the rest of the experimentation, because of the number of replies analyzed. We can see that there is no clear difference between the combined analyzer and the sentiment analyzer, but the latter has shown to be slightly better or equal in all the experiments. This shows that in the case of domains where stress is not normally present, the stress analyzer may only add noise to the results of the calculation of the user state that already considers the output of the sentiment analyzer, thus the system must select to use the combined analyzer or not depending on the domain of application. The stress analyzer by itself continues with a general tendency to be in the 90% to 100% range of concordance.

3.6 Discussion

In this work, we have addressed the topic of sentiment, stress, and combined analysis in the social network domain, and we have discovered that sentiment, stress, and a combination of both found in a written message are good indicators that this polarity, stress level or combined value will propagate to the future messages influenced by the current one. We discovered that the combined analysis presented in this chapter, and using dictionaries of terms for performing aspect-based sentiment and stress analysis works well at least in the domains where stress is present, slightly less well than the sentiment analysis alone in domains where stress is not normally present, like in the Star Wars corpus. For integrating the analyzers as a helping or aiding system for users in an SNS, we designed a MAS that incorporates agents for the sentiment and stress analysis and a novel combined analysis. This system will be able to analyze the sentiment polarity and the stress levels in the data that a user post in an SNS, to perform a combination of the two mentioned analysis, and to decide whether to advise or not the user depending on those values and the concrete case.

In the next chapter, we discuss analysis with more data, using datasets from a public SNS and from a private one, generated in a laboratory with our system active. Additionally, we will introduce new analyzers performing sentiment and stress analysis and a new version of combined analyzer, and will explore the experimentation with public SNS data for discovering which analyzer is more suitable for being able to help the user ever better in his or her social experience. We will integrate our MAS in a SNS and evaluate it with human users in the data gathered from the private SNS in the laboratory.

CHAPTER 4

Sentiment, stress and combined analyses using ANNs

4.1 Introduction

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 chapter, we present a Multi-Agent System (MAS) for assessing guiding users in Social Network Sites (SNSs) 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 [13], where the analyzers were built using a Bayesian classifier. In the current version, we built the analyzers using feed-forward Artificial Neural Networks (ANNs), 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-art accuracies [37]. In [13], 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 improvement against the others that improved the usefulness of the system, in this chapter, 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 chapter is twofold. On the one hand, we constructed a new version of our MAS introducing new analyzers using ANNs, and we used our MAS in experiments in a laboratory with a SNS called Pesedia [12] 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-art works, our proposed approach leverages the use of both MAS technologies and ANNs 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-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 in the current state-of-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 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.

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

The rest of the chapter is structured as follows. Section 2 gives a description of the state-of-art works related to the topic of this chapter. 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 presents the next iteration of the system.

4.2 Related Work

Since our goal is to build a MAS with agents that detect the emotional state of users, using different analysis methods to guide them 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 analysis as well as risk prevention and privacy aiding in SNSs. We will also review previous 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 other than the ones presented in this thesis 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.

As mentioned in the previous chapter, 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 [36]. We use aspect-based sentiment analysis in the approach presented in this chapter, to be able to perform fine-grained sentiment analysis. This is explained more extensively in Section 3.

Aspect detection and sentiment classification are the main topics in aspect-based sentiment analysis [37]. 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 [16]; generative models are also used for detecting aspects such as Conditional Random Fields (CRF), which use a varied set of features [17]; non-supervised machine learning techniques are also used (e.g. Latent Dirichlet Allocation or LDA) [58]. 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 [37]. 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 [37]. Finally, hybrid approaches detect aspects and assign sentiment polarities to them simultaneously [37]. Syntax-based methods obtain words associated with sentiment and extract other aspects by exploiting grammatical relations [20]. CRF are used to relate sentiments to aspects by means of extracting information from relations between words [21].

With regard to aiding privacy in SNSs, in [65], 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 [51]. 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 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 SNSs 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 [43], 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 [44]. The ontology is constructed by monitoring user behavior, and later used for collaborative filtering recommendation, by computing inter-ontology similarities; in [66] 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 [67]. Moreover, other works propose the use of ANNs in MAS architectures for solving problems. In [45], 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 [68], 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 the 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 [46] for supporting business-to-customer (B2C) 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 [69], 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 [70] 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 [71], 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 the 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 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 analyses 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.

4.3 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 [62] 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 [72] 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 4.1. The agents in charge of the analyses and feedback generation of the MAS are explained in more detail in the following sections.

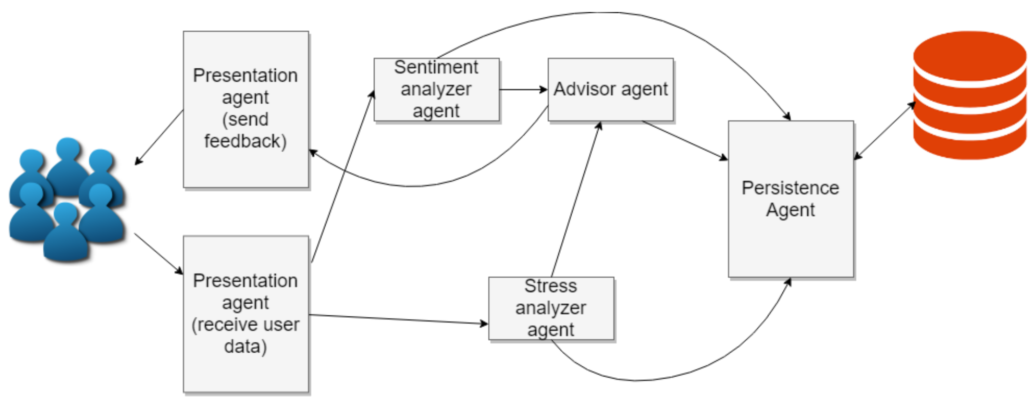


Figure 4.1: Architecture of the MAS

4.3.1 Sentiment analyzer agent

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

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

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 [37] that supervised machine learning techniques have been shown to perform at state-of-art accuracies in the aspect-based sentiment analysis task. The architecture of the network, which is explained below, is shown in figure 4.2.

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 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 [73]. 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 Pleasure, Arousal, and Dominance (PAD) temperament model [74] (Happy, Bored, Relaxed, Anxious, and Angry) and also labeled with a

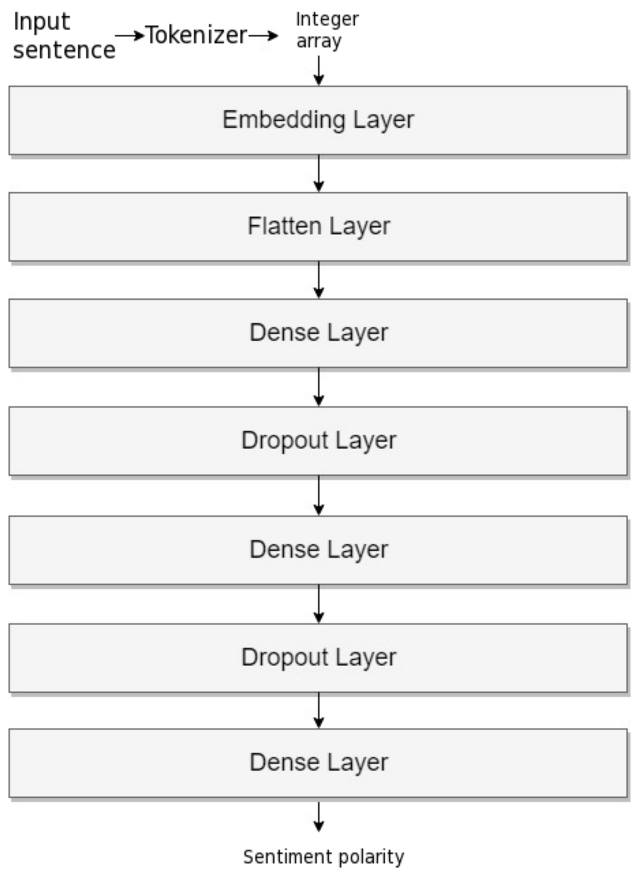


Figure 4.2: Architecture of the ANN for the sentiment analyzer agent

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-art, supervised machine learning, aspect-based sentiment analysis [37], 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).

4.3.2 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 4.3.

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

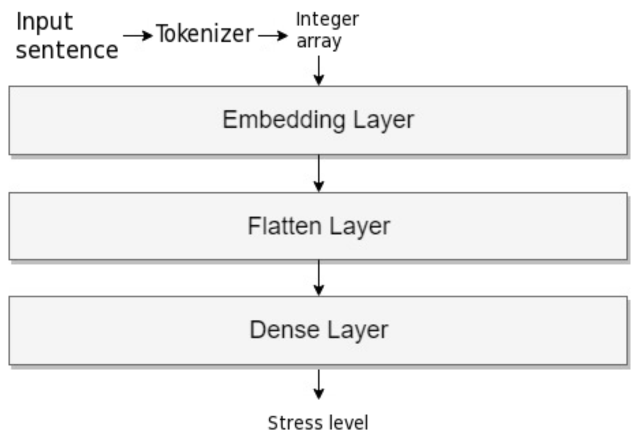


Figure 4.3: Architecture of the ANN for the stress analyzer agent

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-art, supervised machine learning, aspect-based sentiment analysis [37], it can be observed that this classifier achieved state-of-art accuracies.

4.3.3 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 4.4. 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.

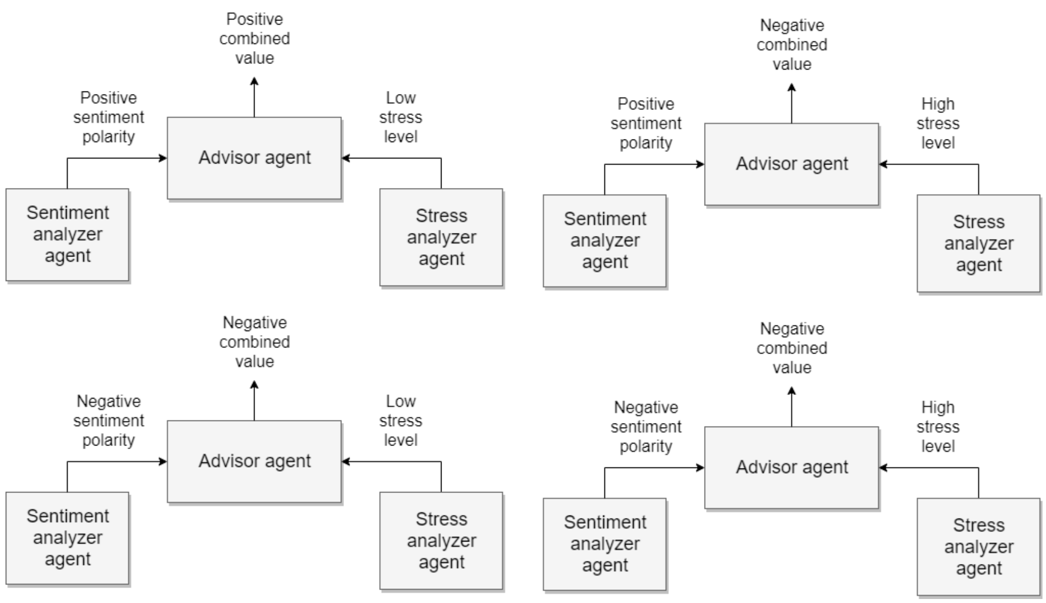


Figure 4.4: Combined analysis integrated in the advisor agent labeling process

4.3.4 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. 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.

4.4 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).

[3]https://elgg.org/

- *negativeMsgs*: Number of messages from the group of people being analyzed that are detected as negative by the system (combined analysis).
- *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 4.1. 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 4.2. 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.

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

Table 4.1: 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 4.2: Comparison of the different analyses in all of the data generated during the experiments

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

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

Table 4.3: 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 %

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

The summarized results of the surveys are presented in table 4.3. 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 %).

4.5 Experimentation with data from Twitter.com

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.com 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.com 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 4.4 for the experiments with the 'or' version of the combined analysis and in table 4.7 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 4.5 for the experiments with the 'or' version of the combined analysis and in table 4.8 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 case of the Podemos corpus. The results for these experiments are shown in table 4.6 for the experiments with the 'or' version of the combined analysis and in table 4.9 for the 'and' version.

Table 4.4: 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

Table 4.5: 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

Table 4.6: 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

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 4.5, 4.6, and 4.7, respectively. For the experiments using the 'and' version of the combined analysis, the results are shown in figures 4.8, 4.9, and 4.10. 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.

Table 4.7: 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

Table 4.8: 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

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

Table 4.9: 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

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 4.10, 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.

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 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 of about 1.5%. The combined analysis in the 'or' version performed worse than the former ones, with this difference being sig-

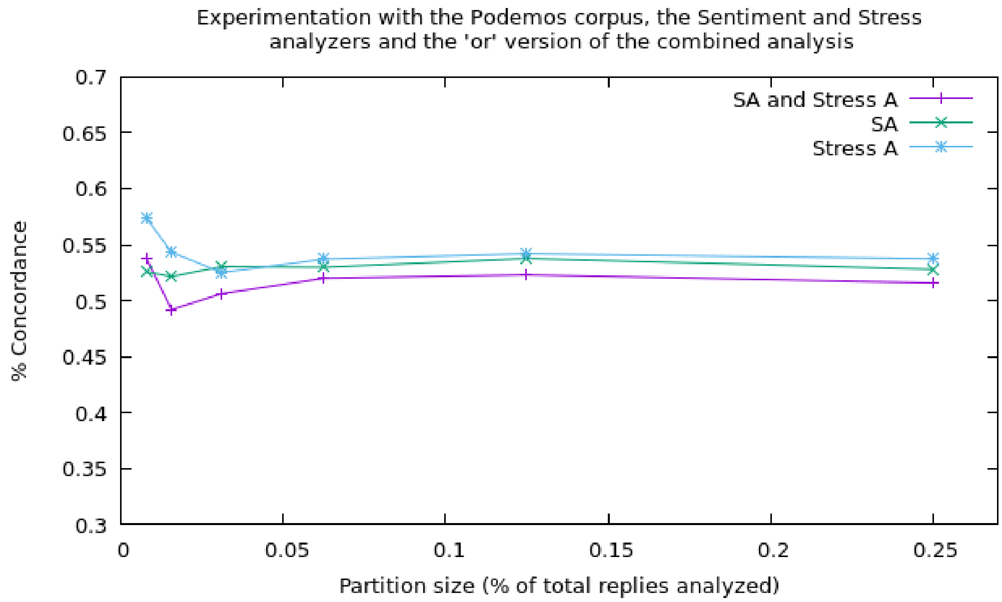


Figure 4.5: Results of the experiments with the Podemos corpus for the sentiment and stress analyzers and the 'or' combined analysis

Table 4.10: Results of t-tests 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

nificant 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

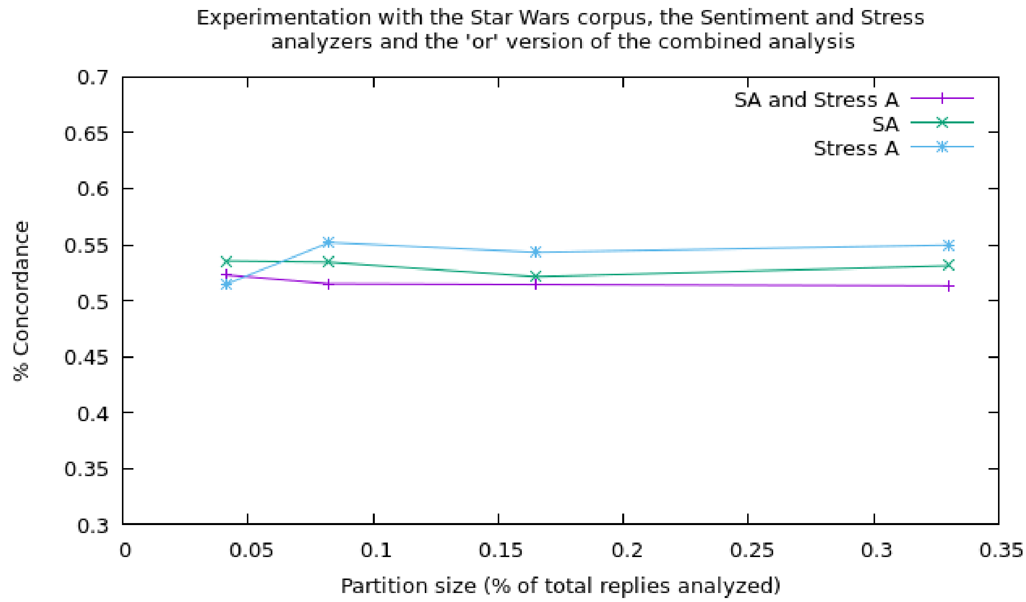


Figure 4.6: Results of the experiments with the Star Wars corpus for the sentiment and stress analyzers and the 'or' combined analysis

that propagates more to the replies, even when 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 sentiment and stress analyzers. The difference between the state detected by the different analyses was again significant according to the respective t-tests. 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.

The results of the t-tests for the Star Wars corpus show that only the

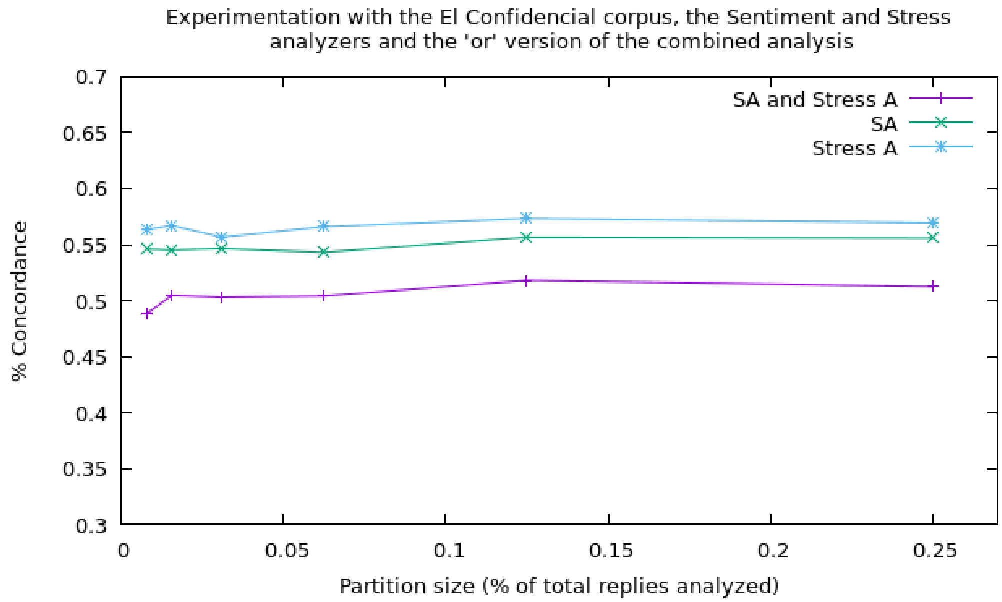


Figure 4.7: Results of the experiments with the El Confidencial corpus for the sentiment and stress analyzers and the 'or' combined analysis

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

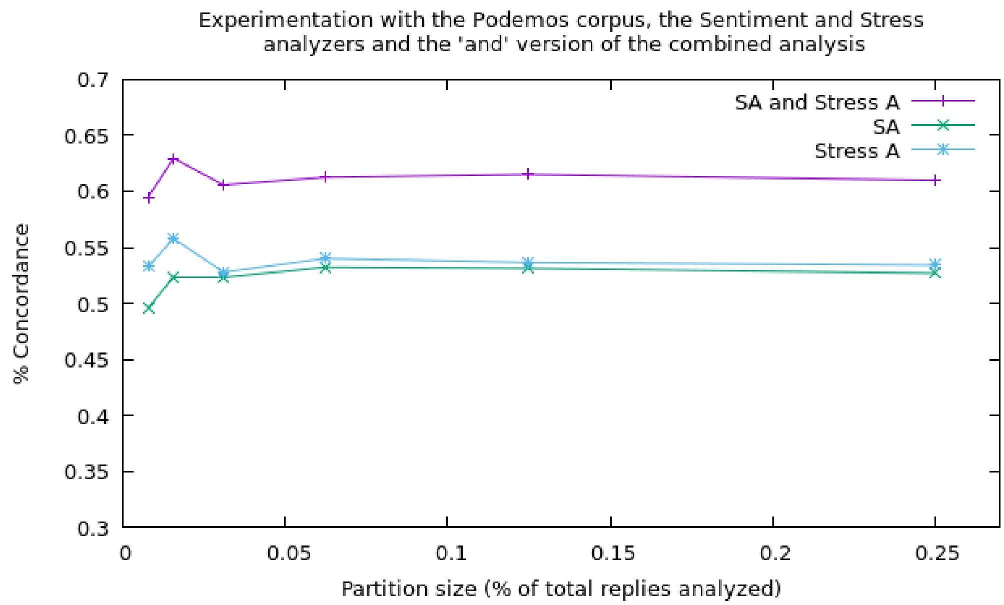


Figure 4.8: Results of the experiments with the Podemos corpus for the sentiment and stress analyzers and the 'and' combined analysis

(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; 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

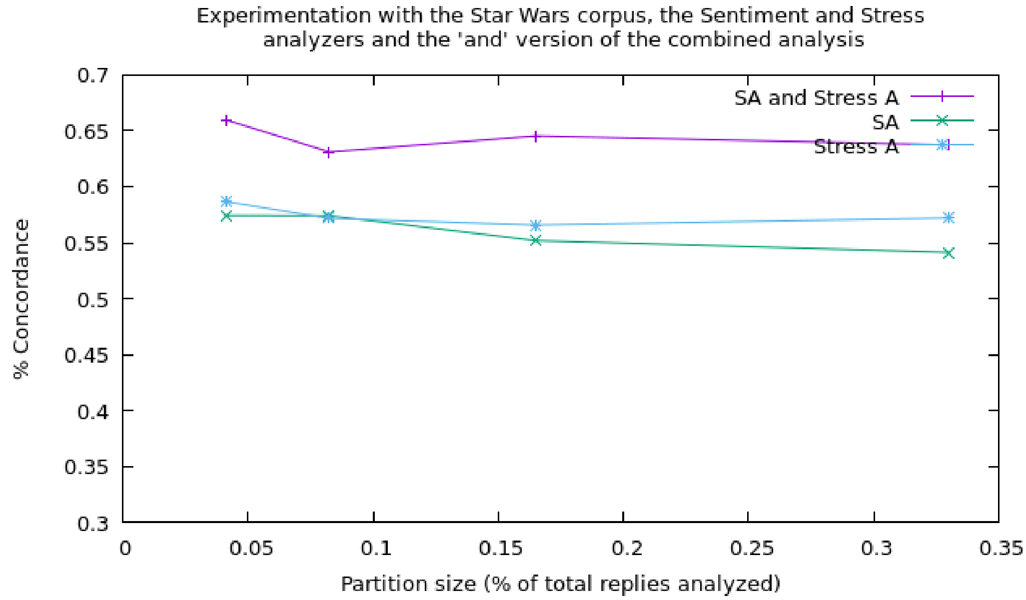


Figure 4.9: Results of the experiments with the Star Wars corpus for the sentiment and stress analyzers and the 'and' combined analysis

'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.

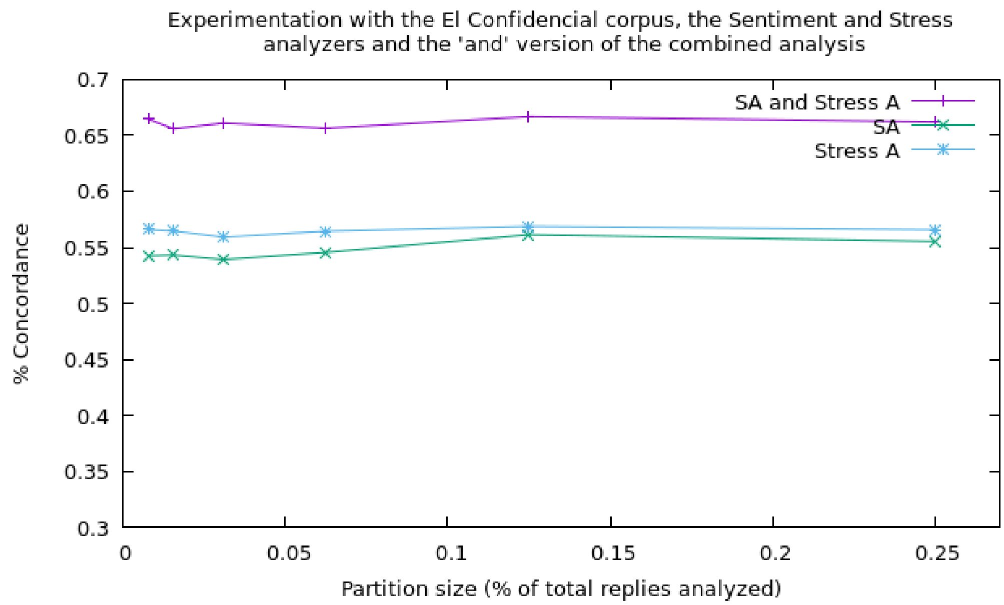


Figure 4.10: Results of the experiments with the El Confidencial corpus for the sentiment and stress analyzers and the 'and' combined analysis

4.6 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.

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.

In the next chapter, we develop new agents that are capable of new types of analyses using other sources of information (typing patterns), and we elaborate experiments performed to discover what analyses work best at detecting user states that propagate more to the other users in the network. We also show a new advisor agent created with a rule-based system that employs the best performing analyses, and considers different situations of the system to use one or other for generating feedback and advice for users navigating in on-line social environments.

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

CHAPTER 5

Keystroke dynamics analysis for sentiment and stress detection

5.1 Introduction

Regarding the influence of the emotional state in the decision-making process, and following the general idea of aiding decision making of users in Social

Network Sites (SNSs) through analyzing their emotional state and stress levels, in previous chapters we presented a Multi-Agent System (MAS) used for analyzing the data of users in a SNS. The analyses in the MAS are used to detect the emotional state and stress level of users and guide them based on this information. In this MAS only sentiment and stress analysis on text data was being performed. Nevertheless, in the literature, we find that keystroke dynamics have been used to successfully build models that detect the sentiment state of a person [75]. Keystroke dynamics are also a non-invasive way of gathering data from users that can be the input to models able to detect sentiment and stress levels. For these reasons, we consider that it is a way to improve the system that we presented previously in [13, 14]. For this purpose, we designed, implemented, and integrated two new analyzers into a new version of our system, which are used to perform sentiment and stress analysis with keystrokes dynamics data, respectively. The analyses of data that are performed by the new analyzers use Artificial Neural Networks (ANNs) to improve the classification accuracy of the system since Machine Learning techniques have achieved state-of-art accuracies in the aspect-based sentiment analysis task [37]. Moreover, a combined version of both sentiment and stress analysis has been proposed for text data and keystroke dynamics data.

The contributions of the present work are then, the design and implementation of agents capable of recognizing sentiment and stress in keystroke dynamics data, and their integration into a MAS that uses this information for guiding users in SNSs. This process has the goal of achieving better recognition of negative states that could produce negative outcomes in SNSs, so the system can better prevent them. For being able to validate our proposal and to check whether there is a better recognition of negative states that propagate in the network, we performed experiments with data from our private SNS Pesedia [12]. In these experiments, we checked if the analyzers are able to detect a state that is also found in replies to messages where it is detected (which we call propagation of the state in the network). We compare the propagation of the state detected with keystroke dynamics analyzers to the state detected by the other analyzers that use text data. Finally, we compare different combined analyses to the analyses that use only one data analysis. Pesedia is a SNS used by young people, both male and female, with ages compressed in the 12 to 15 years old range, that is used in our laboratory experiments and to gather data. It is a SNS made using the social networking engine Elgg[3]. This SNS is built using plug-ins that add functionalities to a base generic networking site, that is provided by the Elgg engine.

Employing the analyzers implemented, the system is able to help prevent contact risks, by warning the user before posting a message with negative sentiment polarity or high-stress level, that could attract unwanted and harmful contacts. The system could also avoid getting publications with negative polarity or high-stress level, which could appear as an effect of the user posting while having negative sentiment or high-stress level. Finally, since the system aims to prevent publications with either negative polarity or high-stress level, it could prevent users from sharing personal information, thus helping prevent the commercial risk. As a matter of example, this could happen if a user that is feeling stressed gets asked for personal information, and due to their high level of stress, they post the information without double-thinking it.

The rest of the chapter is structured as follows. Section 2 reviews state-of-art works related to the topic of this chapter. Section 3 describes our proposed MAS for user guiding in on-line social environments. Section 4 describes the experiments performed with our SNS Pesedia and the new analyzers. Finally, Section 5 exposes conclusions extracted and connects with the next step in the system.

5.2 Related work

In this section, a review is performed on state-of-art works related to keystroke dynamics analysis for building models for sentiment and stress detection, and the influence of emotion and stress on keystroke dynamics, to assess the reason whether it is suitable to add to our system analyzers that use keystroke dynamics as input to detect sentiment and stress. We will also review works about multi-modal sentiment state detection, where various inputs are used for the detection of the user sentiment, which goes in line with the proposed new MAS in this work. Moreover, apart from the one presented in this thesis, to the best of our knowledge, there is no such system that uses a combination of sentiment and stress analysis from text and keystroke dynamics for

[1]https://elgg.org/

discovering if a message could generate a negative repercussion in a SNS and warns the user in the moment of posting a message.

Keystroke dynamics data has been used in the literature to predict the sentiment of the person writing a text. In [41] a group of subjects was asked to type numbers after hearing each of the International Affective Digitized Sounds 2nd edition (IADS-2) [42] and the keystroke dynamics were recorded. They found evidence through statistical analysis that supports that keystroke duration and latency are influenced by arousal. Keystroke dynamics have been studied for discovering the effect of emotion on keystroke data, but they have also been used to build different emotion detection models. In [75] a group of people was asked to type and label the text typed with their emotional state. Then, classifiers for different emotional states that use keystroke dynamics data as input were successfully built, reaching an accuracy of 77.4 % to 87.8 % for the confidence, hesitance, nervousness, relaxation, sadness, and tiredness classifiers.

In regard to the relation between keystroke dynamics and stress levels, in [76] authors gathered keystroke dynamics data from a group of people in two separate scenarios (with normal conditions and under stress). They discovered that about half the keystroke parameters change significantly (after performing the corresponding t-test) from the data of the scenario on normal conditions to the one when the subjects were influenced by stress. Additionally, in [35] authors successfully built different machine learning models that detected cognitive and physical stress using keystroke dynamics features (decision tree, Support Vector Machines (SVMs), ANNs, k-nearest neighbor and AdaBoost).

Multimodal sentiment analysis has started gaining stronger attention from researchers recently. There are three main approaches for assessing multimodal sentiment analysis, which are early, intermediate, and late fusion [77]. Early fusion combines different data sources into a single feature vector. As an example of early fusion, in [78] authors extracted features from audio, video, and text and later fuse them with a multiple kernel learning classifier. Intermediate fusion is performed fusing the data in the intermediate layers of the model itself (e.g. in the intermediate layers of an ANN), and late fusion is the process of combining the outputs of different sentiment classifiers, trained with different modalities of data for giving a final decision of sentiment classification. In [77], three different models are presented, two unimodal models for sentiment classification, using deep Convolutional Neural Networks (CNN) and image data, and other that employs a Long-Short Term Memory network (LSTM) and text data, respectively. The third model combines the output visual features from the CNNs and text features from the LSTM before feeding a fully connected layer with the combination for giving a sentiment classification, which is an example of intermediate fusion. Authors also created a framework for late fusion, where they take into account the output of the three models presented for giving a final sentiment classification. As has been shown, different strategies with unimodal and multimodal data have been employed in the literature for sentiment analysis. Nevertheless, to the best of our knowledge, there is not an approximation that performs sentiment and stress classification using late fusion of text analysis and keystroke dynamics analysis, which are two non-intrusive data modalities. In this work, we propose this form of late fusion, which will be more extensively detailed in the next section.

5.3 System description

In this section, the new analyzers capable of performing sentiment and stress analysis on keystroke dynamics data will be presented, and also their implementation and evaluation will be shown. Moreover, we show the final architecture of our enhanced MAS after the integration of the new agents and explain the process of user guiding, that employs the advisor agent.

We designed agents that can perform sentiment analysis and stress analysis on keystroke dynamics data, and we implemented and integrated them into a MAS that contains agents capable of performing sentiment and stress analysis on text data, initially presented in [14]. This MAS guides users by analyzing their data when they post a message on a SNS and generating feedback when necessary to prevent potential negative outcomes. The MAS is built using the SPADE Multi-Agent platform [62], and every agent performs a different role in the system. There are agents that retrieve data from users in a SNS, and also give back the feedback of the MAS to the users, other agents are in charge of the analyses and feedback generation. Finally, there is an agent in charge of data storage and retrieval in the MAS. A general view of the proposed extended MAS architecture can be seen in figure 5.1.

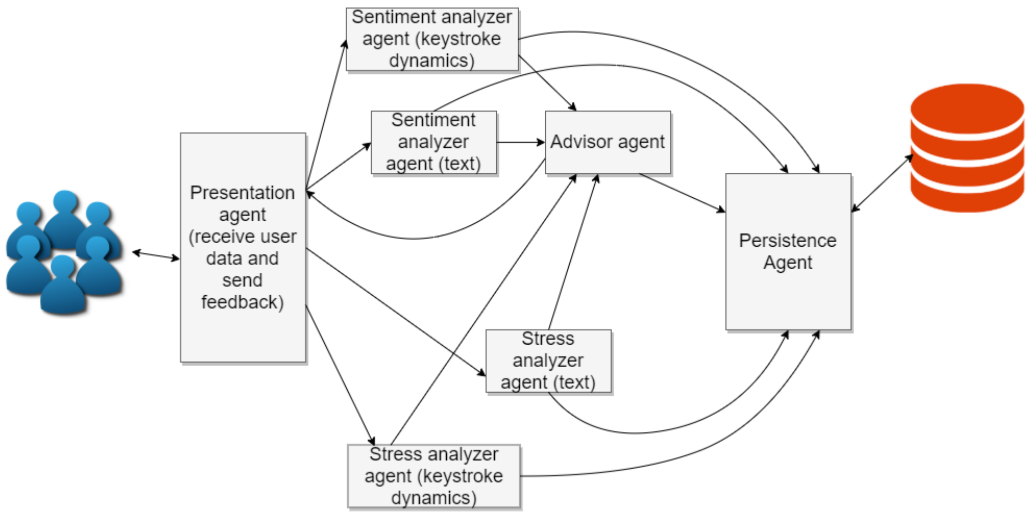


Figure 5.1: Architecture of the MAS

There are four analyzer agents in the MAS, which are the text sentiment analyzer agent, the text stress analyzer agent, the keystroke dynamics sentiment analyzer agent, and finally, the keystroke dynamics stress analyzer agent. The four agents have similarities and differences, but they all take either text messages or keystroke dynamics data as input and give a sentiment polarity or stress level class label as output. The classes for sentiment polarity are 'positive' and 'negative', and for stress levels are 'stressed' or 'no stressed'. We built the different analyzers using a feed-forward ANN trained with Tensorflow[2] version 1.8.0 and Keras[3] version 2.2.0 in the language Python, in its version 3.5.2. For training the ANNs for the analyzers, two datasets were constructed, made with short text messages and with keystroke dynamics samples of users typing messages, respectively. The samples in the datasets were labeled with an emotion label from a set of five emotions inspired in the Pleasure, Arousal, and Dominance (PAD) temperament model [74] (Happy, Bored, Relaxed, Anxious and Angry), and also labeled with a label of 'stress' or 'no stress'. The dataset was built by users of our SNS Pesedia, that as stated before were both male and female, and had ages compressed in the 12 to 15 years old range. It was made using self-report of the users that wrote text messages in the SNS. This labeling process was not mandatory, thus, only the messages that were labeled were inserted into the dataset. Moreover, for being able to train ANNs that consider only two sentiment classes a mapping was made from five emotions to 'negative sentiment' and 'positive sentiment', based on the values of pleasure, arousal, and dominance of these five emotions in the PAD temperament model. The mapping is the following:

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

The functionality of the individual agents in the MAS is the following:

- Presentation agent: This agent receives data from users navigating in a SNS through certain widgets that are used by users to interact with the system. Then, the agent sends this data of text and keystroke dynamics to the analyzer agents. It also receives the feedback generated by the advisor agent and sends it back to users navigating the SNS.
- Sentiment analyzer (text data): This agent computes a sentiment polarity (positive or negative), using text data.
- Stress analyzer (text data): This agent computes a stress level (low or high), using text data.
- Sentiment analyzer (keystroke dynamics data): This agent computes a sentiment polarity (positive or negative), using keystroke dynamics data.
- Stress analyzer (keystroke dynamics data): This agent computes a stress level (low or high), using keystroke dynamics data.
- Advisor agent: The advisor agent calculates the combined analysis from the outputs of the four analyzer agents, and generates feedback for the users if the message is deemed negative. The final definition of the process used for generating the feedback is extracted after the conclusions reached in the experiments with data from our SNS Pesedia, which are both shown in the next section.

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

- Persistence agent: This agent receives data from the analyzer agents and feedback generated and stores it in the database of the MAS.

The process of the MAS starts when a message is being written in the SNS, so the data of the text and keystroke dynamics is sent to the MAS, which calculates the values of sentiment and stress level using both sources of data. When those analyses are performed, the information of predicted sentiment polarity and stress level is sent to the advisor agent, which performs two tasks: the first task is to perform the combined analysis using sentiment and stress levels from the analyses on text data, and on keystroke dynamics data. The other task is the generation of a warning for the user in case the message is deemed as negative by the information generated in the analyses. If the warning is generated, then it is sent to the presentation agent, in charge of sending it as feedback to the SNS, and it is also sent to the persistence agent to store it in the database.

5.3.1 Keystroke dynamics analyzer agents

In this section, the two new agents that perform sentiment and stress analysis on keystroke dynamics data will be presented, and their design and implementation explained, with information about the training of the machine learning models used. Regarding the keystroke dynamics sentiment analyzer agent, it has been trained using keystroke dynamics data from the keystroke dynamics Pesedia self-report dataset mentioned in this section, and the sentiment labels in the dataset are used during the training of the ANN. The architecture of this neural network can be seen in figure 5.2. The collected keystroke dynamics data contains information about the text typing patterns of users, which include text typing speed and character frequency. Text typing speed features included averaged latency for different features, which are listed in table 5.1. For the case of character frequency features, the selected ones are the frequency of pulsation of certain keys, which are listed in table 5.1 as well. Several of the features selected are commonly used in keystroke dynamics analysis (e.g. the interval of time between releasing a key and pressing another, the interval for typing a key sequence, the interval between subsequent key presses, dwell time), and digraph and trigraph features apply these concepts to two-key and three-key sequences [76]. The selected typing speed features are the following:

- key press: Measures the average key press time of users or dwell time (the time a key is pressed), which is included for being able to detect variations on general key dwelling speed that might be caused by sentiment polarities or stress level variations.

- key release and press interval: Measures the average interval of time which takes a user to release a key and press another one. It serves the purpose of detecting variations of key input speed on a different action that the key press. In this case, the interval between releasing a key and pressing another, which denotes de time in which the user starts inputting different information after finishing one.

- key press and second press interval: Similar to the previous one, this time the average interval between two key presses is measured. This interval measures the time that the user uses since they start inputting one information and then, after it is done, start to input another.

- key release and second release interval: Measures the average interval between two key releases, which represents the time that the user spends since some information has been inputted and another one gets inputted as well.

- key press related to digraphs: Since digraph features are timing characteristics for two-key sequences, this feature measures the key press timing related to sequences of two keys, that is, the average key press timing in the digraphs detected at the text, using a list of common digraphs for detecting them. This feature captures timing information that is associated with commonly typed digraphs, which might offer useful information in addition to the previously presented features.

- key release and press interval related to digraphs: The feature that measures average release and press interval for common digraphs.

- key press related to trigraphs: Average key press timing for common trigraphs.

- key release and press interval related to trigraphs: Average release and press interval for common trigraphs.

- digraph typing: averaged value of total time for inputting a digraph.

- trigraph typing: analog to the previous feature but for the case of trigraphs.

- general typing speed: The feature that represents average typing speed.

The selected character frequency features, do not describe or represent keystroke timing information, but instead, they aim to represent user behaviors like corrections (delete, backspace), moving between parts of the webpage (page up, page down, home, end, key up, key down, key left, key right), which might be affected by the sentiment polarity or stress levels. Four commonly used keys are also used as frequency of pulsation features, completing the set of frequency features, which are enter, space bar, shift, and caps lock. Since they are commonly used keys, the differences in the frequency of pulsation might be informative for the machine learning model for predicting sentiment polarities and stress levels, since these states of the user could affect the frequency in which they are used.

The feature vectors, which are fed to the ANN shown in figure 5.2, are vectors of floating-point numbers, corresponding to the presented text typing speed features, followed by the key frequency features, in the order in which they are shown in table 5.1.

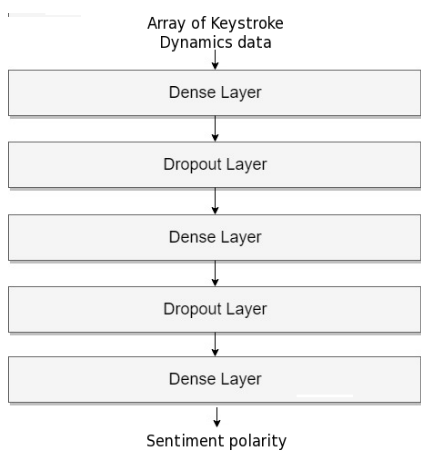


Figure 5.2: Architecture of the ANN for the sentiment analyzer agent with keystroke dynamics data

This data was collected in our SNS Pesedia during a period of one month, and the users were able to perform self-report on their emotional state and stress level, adding finally to the dataset only the samples in which the report of the user state was actually done. This data was later processed with

Table 5.1: Text typing speed and key frequency features used to train the models

Text typing speed features	***Key frequency features***
key press	enter
key release and press interval	space bar
key press and second press interval	back space
key release and second release interval	delete
key press related to digraphs	key up
key release and press interval related to digraphs	key down
key press related to trigraphs	key left
key release and press interval related to trigraphs	key right
digraph typing	shift
trigraph typing	home
general typing speed	end
	page up
	page down
	caps lock

a python script to generate a file that could be used to conveniently feed the ANNs with arrays with the data on the different characteristics, and for eliminating any potential missing data to ensure correctness before starting the training and validation process of the models. The file contains a total of 12313 data samples. For training the models used in the final system, 10% of the data samples were used for training, and a 90% for the validation of the models.

In the architecture of this ANN, the array containing the keystroke dynamics data in the form of floating-point numbers is fed directly to the first dense layer. In figure 5.2 we can see that there are three dense layers in the architecture of the ANN, with two dropout layers placed between the dense layers, acting as a regularization mechanism. The dropout rate of 0.25 in both dropout layers and the architecture of the network were both adjusted experimentally aiming for better accuracy, and for a confusion matrix that showed an equilibrated distribution between the two classes. Several other parameters of the ANN were also adjusted experimentally, which were the sigmoid activation function being used in the dense layers, the inputs, outputs, and neurons of each layer, using binary crossentropy as the loss function for training and an Adam optimizer [73]. The inputs in the dense layers are, in order 25, 64, and 64, with outputs and neurons 64 in all three except in the final dense layer, which is 2. The input vector is the array with the selected features explained in this section, the output 2 in the final dense layer represents the two possible classes (negative and positive sentiment polarity), and finally, the rest were adjusted experimentally as explained before. The dropout layers have the same input, output, and neurons of 64 since they are connected between dense layers, acting as a regularization mechanism. In the validation process, a 61 % of accuracy was reached, which is lower than the accuracies achieved in state-of-art aspect-based sentiment analysis using supervised machine learning (68.0 % to 77.2%) [37], and is on the low side of the accuracies of techniques for discriminating between different affective or valence states using keystroke dynamics (57% to 95.6%) [79]. This could be caused by the dataset of keystroke dynamics data, as it is not a very big dataset and is made using self-report by young people.

In the case of the Keystroke dynamics stress analyzer agent, we find the same ANN architecture as the previous agent, as it was found experimentally to be best for the model accuracy. Again, the keystroke data in the dataset was used for the training, but this time using the stress labels to train the model. Finally, in the validation process, a 64 % of accuracy was reached this time, which is better than the case of the keystroke dynamics sentiment analyzer agent, and is higher than the accuracies found by using different machine learning methods for stress strength detection as shown in [6], but is lower than the accuracy of 75% reported by [35] when detecting stress via a combination of keystroke dynamics and linguistic analysis. There were machine learning methods employed for detecting stress using keystroke dynamics with user personalized models (detection of stress with models trained to identify stress in a concrete user), that reached nearly 90% accuracy [79], and 96.76% to 99.5% accuracy detecting stress states based on the time of the day [80]. However, since these methods are based on user personalized models, they are not comparable to our approach, which is based on user-independent models (we use the same models to predict stress in all the users instead of personalized models).

5.4 Experiments with data from the SNS Pesedia

We conducted experiments with our SNS Pesedia during a period of one month. During that time, we gathered the text and keystroke dynamics data of the user when he or she made a post on Pesedia, so we could analyze these two sources of information in later experiments. As stated previously, Pesedia was used by children with ages ranging from twelve to fifteen years old. In Pesedia, the users can perform several actions related to social interaction in SNSs. These actions include between others: Post messages in their wall or other walls; send private messages; create groups and invite people; make friends; share content with certain people; making lists of people. In our experiments, we make use of the messages posted on walls and groups and their replies. We also use the concept of propagation. By propagation, we refer to the fact that a detected state by an analyzer on a message is also found in the majority of the replies to the message in which the state is detected. Only direct replies are considered, not replies to replies. This design decision was taken to be able to study the effect of the emotional state detected in a message on its replies, in future works the effect in multiple levels of replies or other messages might be tested.

Our aims in the experimentation are two: On the one hand, to discover if the new analyzers integrated into the system predict a state that propagates more in the network, and to know which analyzer achieves the best propagation in general. In this way, we would be able to know which analyses can be more helpful for the feedback generation to the user, aiming to prevent a negative repercussion on the social network. On the other hand, to be able to know if the combined analysis is able to detect a state of the user that propagates more in the network than the analyses using only text or only keystroke dynamics data.

5.4.1 Metric for the experiments

In this section, the metric that is used in the experiments will be explained using a formula. This metric is Propagation of detected value (PDV). The following terms are later used for the formula of the metric:

- *messages_with_replies*: Total amount of messages that generated replies.

- *messages_with_propagated_state*: Aggregated value of messages with propagated state, which are messages with the same detected state than the most present in their replies.

The following formula describes the calculation of the metric used in the experiments conducted:

$$PDV = \frac{messages\_with\_propagated\_state}{messages\_with\_replies}$$

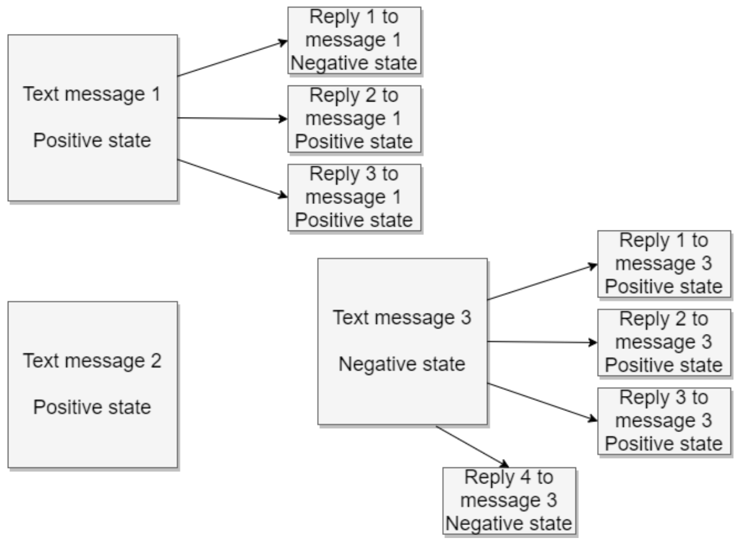


Figure 5.3: Example of messages posted and replies with different polarities

As a matter of example, we propose the following scenario: There are three messages posted in the social network, two of them with replies and one without. All the messages and their replies have been analyzed and a state associated with them by the system. This can be seen in figure 5.3. As can be seen, the first message generated three replies and two of them were positive (making the most present value in the replies positive), so there has been propagation from the original message to the replies since this original message also resulted in positive. For the case of the third message, there is no propagation since there are three replies detected as positive and only one negative, and the original message was detected as negative. In this scenario, messages_with_replies would be computed as two since only two messages have replies. For computing Messages_with_propagated_state, we sum the number of messages that generated replies and propagated the state detected on them to those replies. As has been commented, only one of the messages propagated the state to the replies, therefore the value of Messages_with_propagated_state in this example is one. Finally, the value of PDV for this example would be computed as 0.5.

5.4.2 Plan of the experiments

We used the messages that had or generated replies in the network for our experiment and used the text data and keystroke dynamics data of the user writing those messages. The goal of the experimentation was to compute the propagation of the detected value in the messages in the network, by comparing this value to the most present value detected in the replies of the messages. This propagation was measured for the different analyses that we had available, which are sentiment, stress, and combined analysis using text, and then again sentiment, stress, and combined but using keystroke dynamics data. We also computed the propagation for a combined analysis that used both combined sentiment and stress analysis on text data, and combined sentiment and stress analysis on keystroke dynamics data.

We performed the training of the ANNs that would conduct the different analyses using different partitions of the training dataset mentioned in Section 3 each time. We performed 5 different partitions of the dataset. In this way, we obtained five ANNs trained to detect sentiment and five trained to detect stress levels on text data, and again five ANNs to detect sentiment and five to detect stress levels with keystroke dynamics data, using different partitions of the data for training each one. This process was done for being able to analyze not only differences between analyzers, but also for exploring the differences of ANNs trained with different data. For this purpose, we proceed to perform experiments with the data not used for training (we separated the messages that were used for training from the dataset for conducting these experiments), comparing the ANNs that detect sentiment with the ones that detect stress labels on text data, and the same with the ones that operate with keystroke dynamics data. Giving a total of 25 experiments on text and 25 on keystroke dynamics data, since we compared every pair of text ANNs and every pair of keystroke dynamics ANNs.

Finally, we launched several experiments with the setup presented in this section, changing the threshold of class inference in the ANNs from 0.5 to 0.9. This means that if the probability of a class given as output by the ANN model is greater or equal than the threshold, the result of the analysis is this concrete class, otherwise, it is the other class. We show in figure 5.4 the results of propagation for the analyses on text data, and the combined analyses of text and keystroke dynamics data with thresholds for class detection unaltered. figure 5.5 shows the results of propagation of analyses on keystroke dynamics data and the combined analyses with thresholds unaltered again. Then, for analyzing the effect of altering the threshold on the ANNs, we show the following figures, where the threshold is set to 0.7 (we don't show the other possibilities such as threshold 0.9 since 0.7 was shown experimentally to be the only case where there are bigger differences with the base case with no changes). We show in figure 5.6 the results of the analyses on text data and combined analyses (text and keystroke), while changing the threshold of the ANNs that work with keystroke dynamics data to 0.7. We show the results for the analyses on keystroke dynamics data and the combined analyses applying this same process in figure 5.7. Altering the threshold in ANNs that work with text data to 0.7, we repeated this process and obtained the figures 5.8 and 5.9. Finally, we show in figure 5.10 and figure 5.11 the results altering the threshold to 0.7 in the ANNs that perform sentiment analysis, and in figure 5.12 and figure 5.13 the results altering the threshold to 0.7 in the ANNs that perform stress analysis.

In the figures, the legend represents the following forms of the metric presented in Section 4.1:

- PCOMB_or_text: PDV of the 'or' version of combined sentiment and stress analyzers on text.
- PSEN_text: PDV of the sentiment analyzer on text.
- PSTR_text: PDV of the stress analyzer on text.
- PCOMB_and_text: PDV of the 'and' version of combined sentiment and stress analyzers on text.
- PCOMB_or_ksd: PDV of the 'or' version of combined sentiment and stress analyzers on keystroke dynamics.
- PSEN_ksd: PDV of the sentiment analyzer on keystroke dynamics.
- PSTR_ksd: PDV of the stress analyzer on keystroke dynamics.
- PCOMB_and_ksd: PDV of the 'and' version of combined sentiment and stress analyzers on keystroke dynamics.

- PCOMB_TEXT_OR_KSD_or: 'or' version of combined analysis, using the values resulting from the output of PCOMB_or_text and PCOMB_or_ksd.

- PCOMB_TEXT_OR_KSD_and: 'and' version of combined analysis, using the values resulting from the output of PCOMB_or_text and PCOMB_or_ksd.

- PCOMB_TEXT_AND_KSD_or: 'or' version of combined analysis, using the values resulting from the output of PCOMB_and_text and PCOMB_and_ksd.

- PCOMB_TEXT_AND_KSD_and: 'and' version of combined analysis, using the values resulting from the output of PCOMB_and_text and PCOMB_and_ksd.

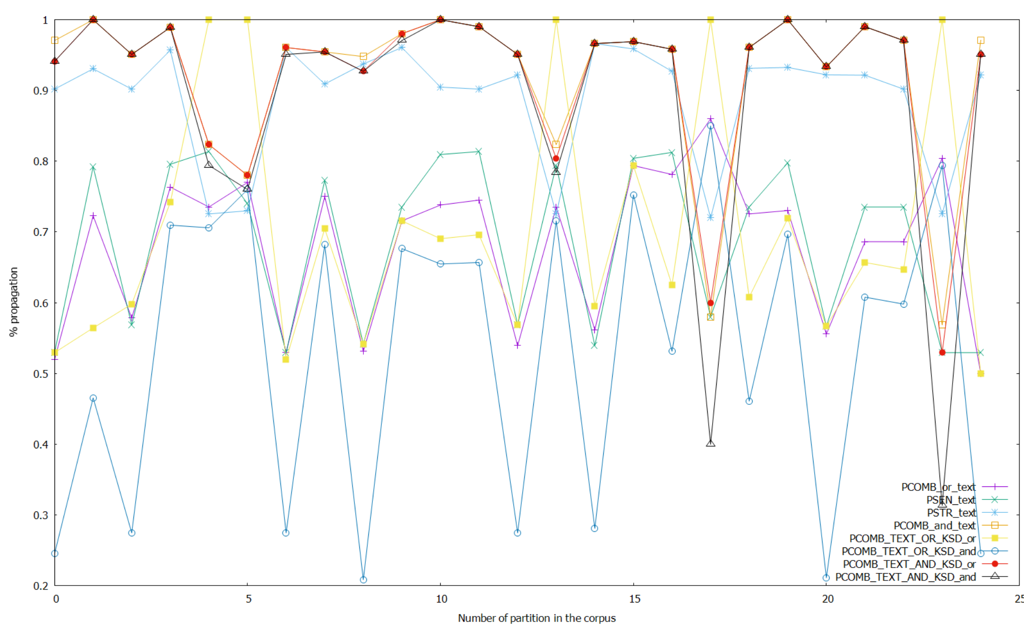


Figure 5.4: Results of the experiments with the sentiment, stress, and combined analyses on text, and with the combined analyses on text and keystroke data with thresholds unaltered

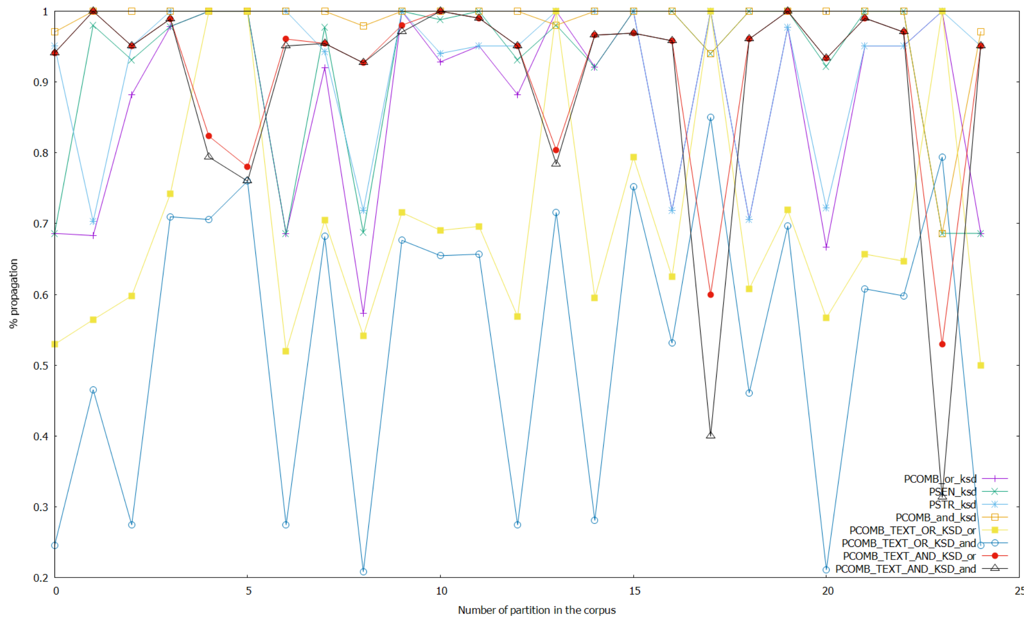


Figure 5.5: Results of the experiments with the sentiment, stress, and combined analyses on keystroke data, and with the combined analyses on text and keystroke data with thresholds unaltered

5.4.3 Results

As table 5.2 shows, in the experiments with both the ANNs trained with text data (embeddings of texts) and the ones trained with data of keystroke dynamics, there is a propagation of the detected state of the user to the replies of the message being analyzed. We can appreciate that there are several differences between experiments with different partitions of the data used on the training and in the experimentation, even when the general trend is to find a propagation of the state. There were cases, that provided a network that was very likely to give as output one of the two classes (negative or positive sentiment for sentiment analysis, or in the case of stress analysis high or low stress level). This was because the data in the partition of the dataset used for the experiment was unbalanced in favor of one of the classes, which resulted in high propagation in the later experiment (since the ANN was very likely to output one of the two classes, resulting in that the comparison between the analysis of the message and the replies would match most of the time). Also, some cases give a poor propagation, since the data in that parti-

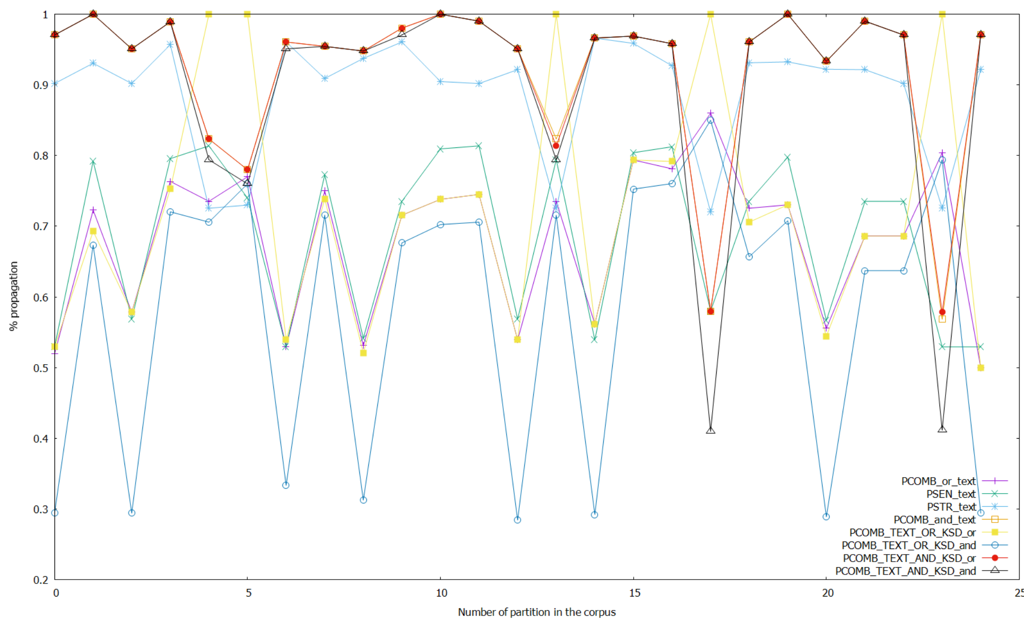


Figure 5.6: Results of the experiments with the sentiment, stress, and combined analyses on text, and with the combined analyses on text and keystroke data with threshold set to 0.7 in the ANNs that work with keystroke data

tion could not provide an ANN that would give satisfying results of accuracy.

The average propagated values for the different analyses are shown in table 5.2. In the experiments without altering the thresholds for class detection, the best-performing analyses in terms of propagation for combined analyses of sentiment and stress that use fusion of text and keystroke data (combining the output of the respective analyzers) are PCOMB_TEXT_AND_KSD_and (0.8951 average propagation) and PCOMB_TEXT_AND_KSD_or (0.9153 average propagation), being the latter better than the former. For combined analyses of sentiment and stress that don't use data fusion, the best-performing analyses were PCOMB_and_text (0.9196 average propagation) and PCOMB_and_ksd (0.9811 average propagation), being the latter better than the former again. As shown in table 5.2 the non-fusion analyses also shown higher average propagation than the analyses using text and keystroke dynamics data. PSEN_ksd, PSTR_ksd are the two non-combined and non-fusion analyses with the highest propagation (0.9193 and 0.9235 average propagation respectively) and PSTR_ksd also has higher propagation than

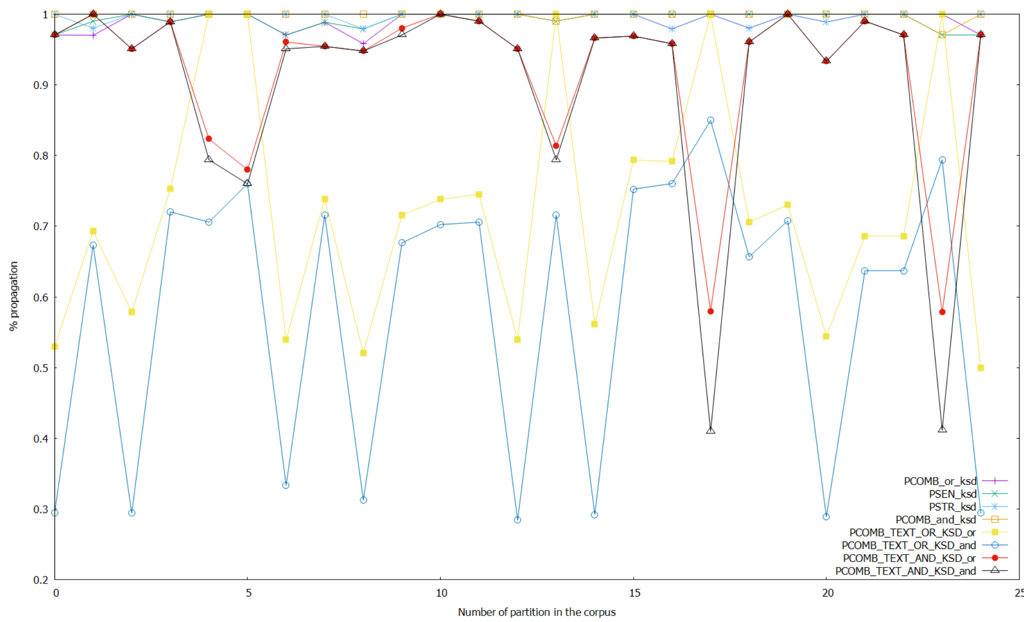


Figure 5.7: Results of the experiments with the sentiment, stress, and combined analyses on keystroke data, and with the combined analyses on text and keystroke data with threshold set to 0.7 in the ANNs that work with keystroke data

PCOMB_and_text but lower than PCOMB_and_ksd. Finally, PCOMB_or_ksd also shown a high result of propagation, but lower than the best performing analyses already discussed in this section. The worst performing analyses were PCOMB_TEXT_OR_KSD_or, PCOMB_TEXT_OR_KSD_and, PCOMB_or_text, and PSEN_text (0.7033, 0.5333, 0.6823, and 0.6868 average propagation respectively), being PCOMB_TEXT_OR_KSD_or better.

Regarding the experimentation altering the threshold for class detection for the ANNs, it can be seen that as a general trend, every analysis affected obtains a better propagation of the state detected upon setting the threshold to a more strict point (e.g. 0.7 like in the results shown on figures 5.4 to 5.13). On the one hand, when altering the threshold of the ANNs that work with keystroke dynamics data setting it to 0.7, it is shown that PCOMB_TEXT_AND_KSD_or (best performing fusion analysis) is able to perform with a very similar propagation than PCOMB_and_text (which is the best analysis using only text data, while analyses using keystroke data

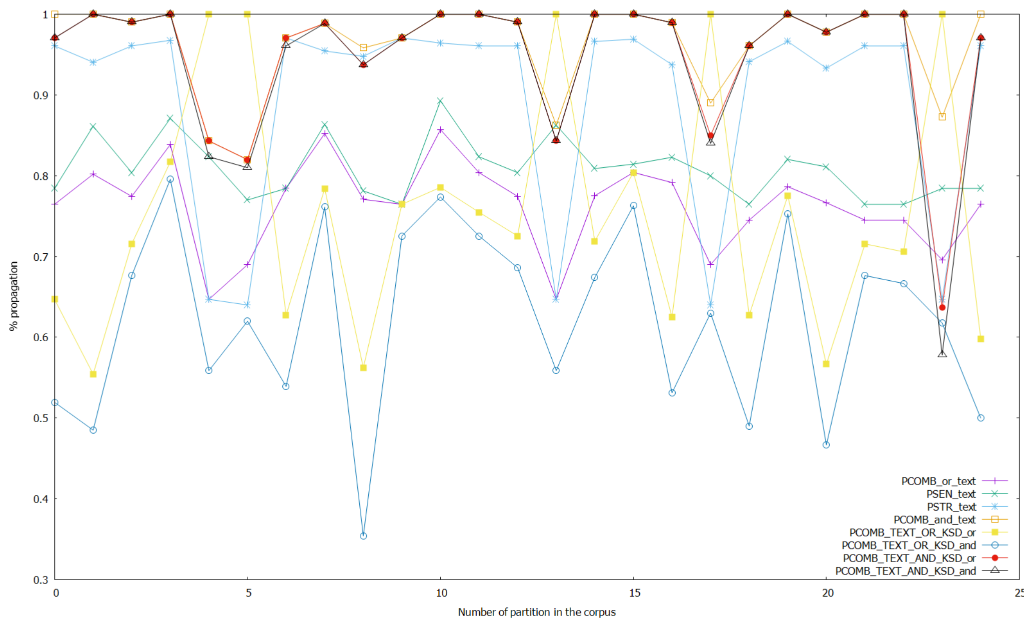


Figure 5.8: Results of the experiments with the sentiment, stress, and combined analyses on text, and with the combined analyses on text and keystroke data with threshold set to 0.7 in the ANNs that work with text data

were close to total propagation because of making the threshold strict for this data). Moreover, making a more strict detection on the ANNs that perform sentiment analysis we obtained a similar case to the one when the thresholds on ANNs that work with keystroke data were altered. PCOMB_TEXT_AND_KSD_or performs similarly than PCOMB_and_text, and we also see that in this case, the combined analyses that use fusion of text and keystroke data perform better in general. On the other hand, if a strict threshold for class detection on the ANNs that work with text data is set, even when it can be seen that the combined analyses that perform fusion of text and keystroke data improve in terms of propagation, this change does not accomplish making these analyses better than the best performing non-fusion analyses (PCOMB_and_text and PCOMB_and_ksd). Finally, setting the threshold for the ANNs that perform stress analysis to strict detection does not have a strong effect on the propagation in general (even when a small improvement from the case without altering thresholds can be seen in general), except in the case of the stress analysis and PCOMB_or_ksd, that improves a 6.14% from the case without altering thresholds, while still performing lower

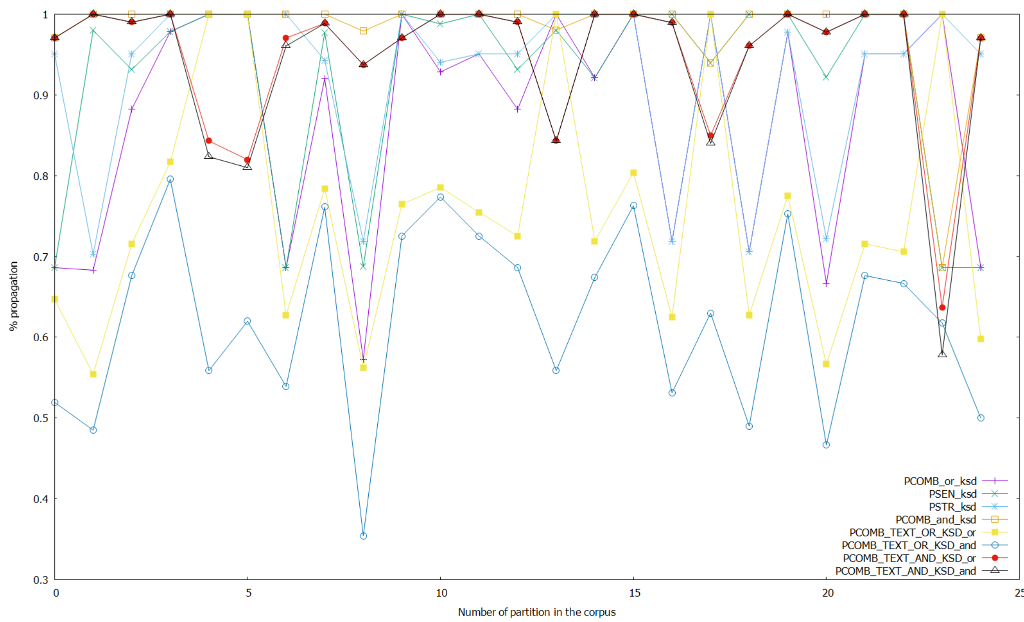


Figure 5.9: Results of the experiments with the sentiment, stress, and combined analyses on keystroke data, and with the combined analyses on text and keystroke data with threshold set to 0.7 in the ANNs that work with text data

than PCOMB_and_ksd. Additionally, this case does not manage to make the combined fusion analyses to be better than the best performing non-fusion analyses.

5.4.4 Reformulation of the advisor agent

Since there has been an addition of new analyzers in the system and experiments performed for knowing which analyzers work best at detecting states of the users that propagate more, and therefore might be more informative for the advisor agent, this agent has been redesigned and its decision-making process for generating feedback updated. The selection of the analyses used in the advisor agent is done because the experiments have shown that as a general trend they are the best at detecting a state that propagates more in the network, and they are shown following. As has been stated in the previous section, the advisor agent accomplishes two different tasks: the calculation of the combined analyses from the data given by the four analyzer

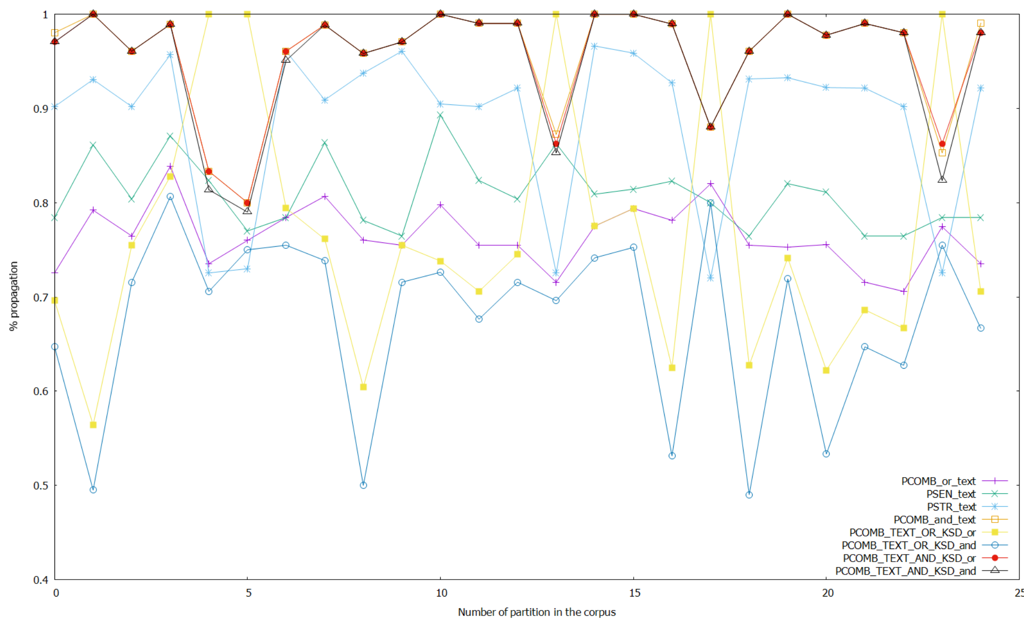


Figure 5.10: Results of the experiments with the sentiment, stress, and combined analyses on text, and with the combined analyses on text and keystroke data with threshold set to 0.7 in the ANNs that perform sentiment analysis

agents, and the generation of feedback to the users interacting in the SNS that generated the message being analyzed.

In accordance with the experiments performed, the first task of the advisor agent has been determined to be the calculation of three different combined analyses, selected from the results of the experiments that aimed to discover which analysis is more informative at detecting a state of the user that propagates more in the SNS. The three combined analyses computed in this agent are the following: PCOMB_and_text, PCOMB_and_ksd, and PCOMB_TEXT_AND_KSD_or. The two combined analyses that use only one data source (PCOMB_and_text and PCOMB_and_ksd) use the 'and' version of combined analysis. Finally, the combined analysis of text and keystroke dynamics uses an 'or' version of combined analysis, taking the output of the text combined analysis and keystroke dynamics combined analysis as input. This selection of analyses is done based on the results of the experiments, as has been shown in figures 5.4 to 5.13.

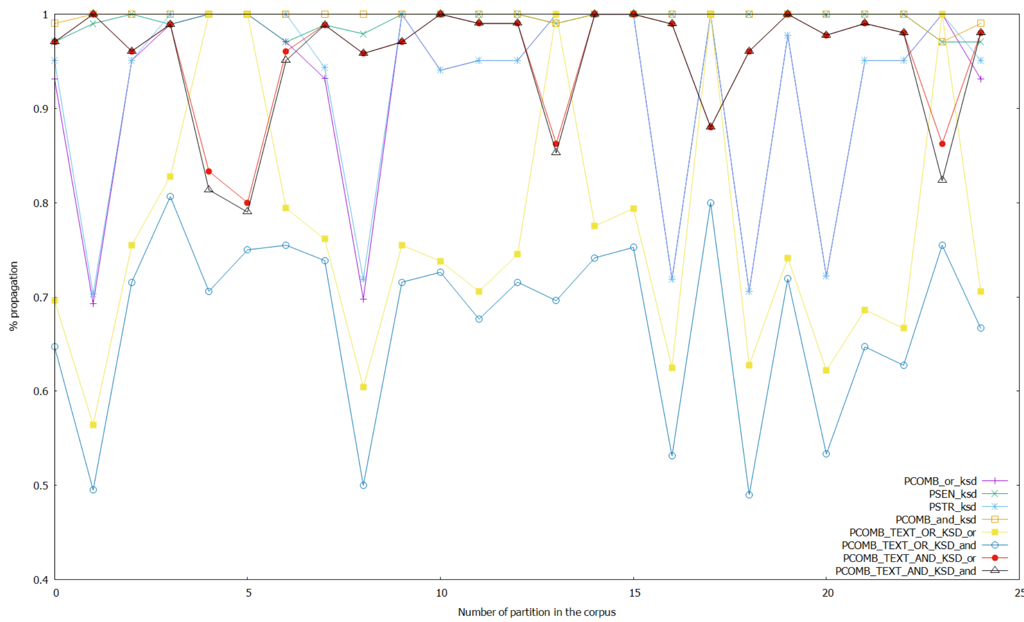


Figure 5.11: Results of the experiments with the sentiment, stress, and combined analyses on keystroke data, and with the combined analyses on text and keystroke data with threshold set to 0.7 in the ANNs that perform sentiment analysis

The second task of this agent is to generate feedback to the user, based on the data generated on the analyses. The rules used for this process are based on the following:

1. As the default case, if there are no issues detecting input, or with data availability or neutral detections, then the warning is generated when a negative message is detected by the text and keystroke dynamics data combined analysis.

2. The tokenizer sometimes is not able to detect any token in the input text, in this case, the feedback to warn the user is generated when the sentiment and stress combined analysis on keystroke dynamics data assigns a negative label to the message.

3. One source of data is not available, being text or keystroke dynamics, so the warning is generated if the combined analysis on the available data detects a negative message.

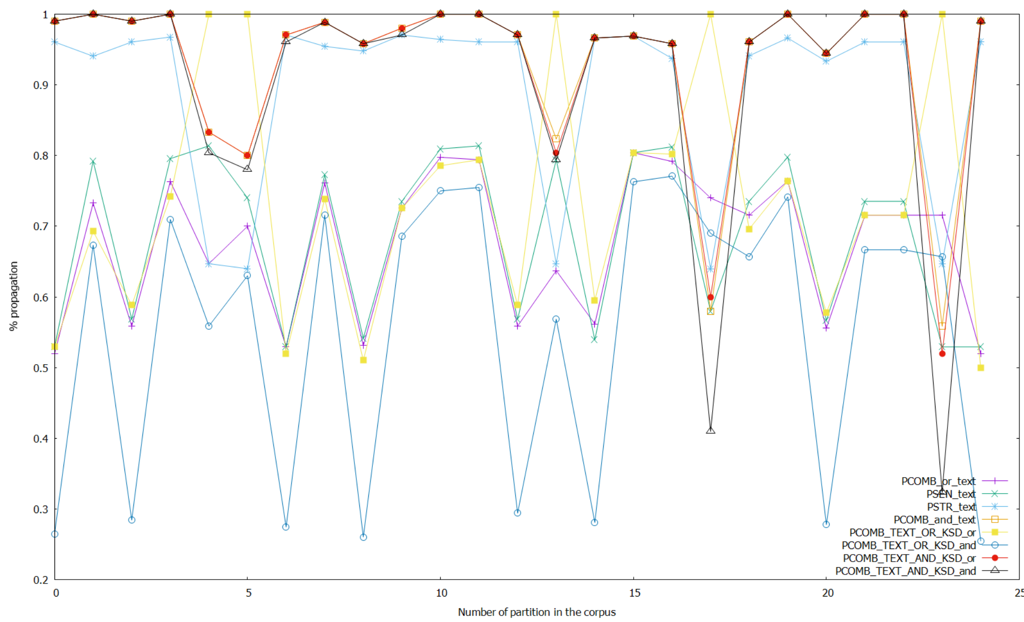


Figure 5.12: Results of the experiments with the sentiment, stress, and combined analyses on text, and with the combined analyses on text and keystroke data with threshold set to 0.7 in the ANNs that perform stress analysis

4. The class probability detected is in the range of 0.5 - 0.525 for both sentiment and stress analysis on one kind of data, meaning that the ANN models for this data source detect a neutral state. In this case, the warning is generated if the combined analysis on the other data source detects a negative message outside of the mentioned range of class probability. If both data sources generated an output with a class probability inside the mentioned range, then the combined analysis of text and keystroke dynamics data is used for the generation of the warning, doing so if a negative message is detected by this combined analysis.

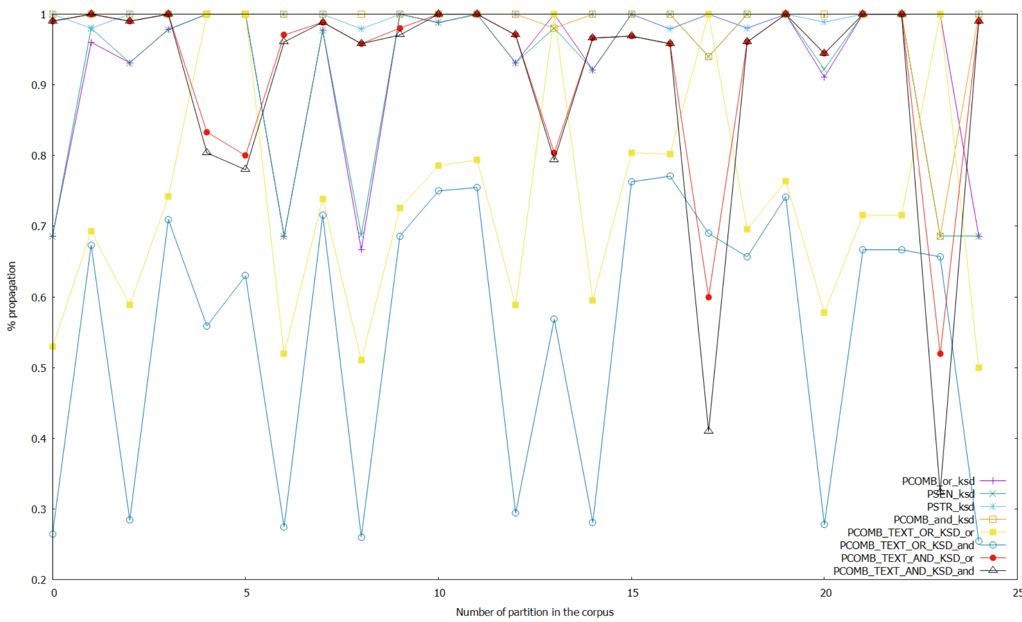


Figure 5.13: Results of the experiments with the sentiment, stress, and combined analyses on keystroke data, and with the combined analyses on text and keystroke data with threshold set to 0.7 in the ANNs that perform stress analysis

Table 5.2: Mean and standard error (in parenthesis) for the different versions of PDV in the experiments: PSTR_text (1), PCOMB_or_text (2), PSEN_text (3), PCOMB_and_text (4), PSTR_ksd (5), PCOMB_or_ksd (6), PSEN_ksd (7), PCOMB_and_ksd (8), PCOMB_TEXT_OR_KSD_or (9), PCOMB_TEXT_OR_KSD_and (10), PCOMB_TEXT_AND_KSD_or (11), and PCOMB_TEXT_AND_KSD_and (12)

Version of PDV	No changes in the thresholds	Threshold 0.7 in sentiment analysis	Threshold 0.7 in stress analysis	Threshold 0.7 in analyses on text	Threshold 0.7 in analyses on keystroke data
1	0.8879 (0.0171)	0.8879 (0.0171)	0.8951 (0.0257)	0.8951 (0.0257)	0.8879 (0.0171)
2	0.6823 (0.0214)	0.7644 (0.0066)	0.6743 (0.0203)	0.7633 (0.0110)	0.6823 (0.0214)
3	0.6868 (0.0238)	0.8093 (0.0074)	0.6868 (0.0238)	0.8093 (0.0074)	0.6868 (0.0238)
4	**0.9196** (0.0237)	**0.9567** (0.0117)	**0.9293** (0.0244)	**0.9634** (0.0113)	**0.9196** (0.0237)
5	**0.9235** (0.0219)	**0.9235** (0.0219)	**0.9963** (0.0015)	**0.9235** (0.0219)	**0.9963** (0.0015)
6	0.8700 (0.0285)	**0.9186** (0.0221)	**0.9314** (0.0229)	0.8700 (0.0285)	**0.9907** (0.0027)
7	**0.9193** (0.0244)	**0.9928** (0.0022)	**0.9193** (0.0244)	**0.9193** (0.0244)	**0.9928** (0.0022)
8	**0.9811** (0.0126)	**0.9976** (0.0013)	**0.9843** (0.0127)	**0.9811** (0.0126)	**0.9984** (0.0012)
9	0.7033 (0.0337)	0.7676 (0.0270)	0.7354 (0.0330)	0.7551 (0.0293)	0.7236 (0.0335)
10	0.5333 (0.0432)	0.6763 (0.0191)	0.5541 (0.0405)	0.6220 (0.0233)	0.5826 (0.0407)
11	**0.9153** (0.0242)	**0.9559** (0.0116)	**0.9278** (0.0251)	**0.9484** (0.0173)	**0.9196** (0.0235)
12	**0.8951** (0.0350)	**0.9523** (0.0129)	**0.9092** (0.0352)	**0.9441** (0.0195)	**0.9026** (0.0323)

5.5 Conclusions and future work

In the present work, we introduced new agents capable of performing sentiment and stress analysis on keystroke dynamics data into a MAS presented in a previous work, for being able to improve the capacity of the MAS when predicting user states that could generate a problem or make risks arise from the social interaction. The MAS guides users on a SNS or other on-line social environment by analyzing the data of a user that posts a message and giving feedback to the user if this message is deemed negative by the MAS agents analyses. We also propose different versions of combined analysis that use both sentiment analysis and stress analysis on text data and on keystroke dynamics data, to be used in the advisor agent of the MAS for the task of generating feedback to the users. For discovering what analyses are more informative to be used in the generation of warnings or feedback, and thus improve the system ability to prevent negative outcomes and risks in a social on-line environment, we integrated our MAS into a private SNS called Pesedia and performed experiments with real users during a period of one month. During this period, we gathered the dataset of text and keystroke dynamics data used for training the machine learning models. We performed a laboratory experiment with the data from the dataset not used for training that aimed at discovering which analyses are able to predict a state of the user that propagates more to the replies of the message analyzed in the network. In this way, we would be able to know what analyses can be considered more informative to be taken into account for warning the users, in terms of their ability to detect a state that may potentially propagate more in the network. We also launched experiments with different setups in the threshold for class detection in the ANNs that perform the analyses.

Regarding the experimentation with the different analyses for discovering which one detects a state that propagates more to the replies, we found that the best analyses are the 'or' version of combined text and keystroke analysis that uses 'and' combinations of sentiment and stress analysis, and the 'and' versions of text combined analysis and keystroke dynamics combined analysis. Analyses on only keystroke dynamics also shown high results of propagation but the combined version of sentiment and stress on keystroke dynamics was better than the non-combined analyses. Moreover, when setting a high threshold for class detection (making the detection process more strict, by selecting one class only with a 70% of probability or more), in the

ANNs that perform analyses on keystroke dynamics data and in the ANNs that perform sentiment analyses, the best analysis that combines text and keystroke data approaches the best single data type analysis. This is not the case when altering the threshold for the ANNs that perform analyses on text data and that perform stress analysis.

The proposed approach, as shown by the experiments can help users navigating in an on-line social environment be aware that the information they post has a chance of generating risks in the network for the users navigating, and thus can help prevent those risks. Nevertheless, the target users in our proposed system are people of young age, and so we built and trained models with data from people of ages ranged between 12 to 15 years old, therefore it is not granted that the performance of the system implemented will be the same when used in on-line social environments with older or more experienced users. Despite this limitation, in [13, 14] we built text data analyzers (which also included combined analyses) that were able to predict sentiment and stress states of the user that propagated to the replies of messages in the SNS Twitter.com, which has a wide range of users, therefore it has been shown that analyzers of text data can be used to this purpose. Moreover, data privacy is a limitation for the proposed system, since it relies on the analyses on user data to generate feedback and potentially prevent risks. Consequently, if the users do not consent to the use and analysis of their data, the system would be unable to work and provide feedback or warnings.

In the following chapter, we exploit the different available data and analyses in the MAS with a Case-Based Reasoning (CBR) module, to be able to generate feedback that is more useful to the user. The CBR uses also information from the context of interactions, such as the topics being talked about in text messages, or the history of detected states by the system in the last set of interactions of the user and the audience of the messages of this user. This new module is expected to improve the current advisor agent, which uses a series of rules to decide which output from the analyzers to use to give or not feedback and warnings to the users. Experiments assessing the error rate of the CBR module with different configurations when populating the case base and update intervals are performed and discussed, and also experiments for discovering if the CBR module is able to detect states of the users that propagate more in the network than individual analyzers, such as the sentiment analyzer on text or the one on keystroke data.

CHAPTER 6

CBR module using data analysis and context information

6.1 Introduction

As presented in previous chapters, the Multi-Agent System (MAS) for guiding users in on-line social environments has evolved, using different techniques and analyses of the user states. In [14] we presented a MAS as a system that collects messages from users that are interacting in a Social Network Site (SNS) or other social environment and computes sentiment, stress, and a combined analysis for potentially generating feedback to the user, as a warning that aimed to avoid future negative repercussions on the social environment. Keystroke Dynamics (KSD) are timing information and frequency of pulsation of keys that can be collected when a user is typing on a keyboard and used as an additional source of information for a data analysis application. In [10] agents performing sentiment and stress analysis on keystroke dynamics data were created and used together with agents that analyze text data for performing sentiment and stress analysis. Different analyzers were proposed, including single modality analysis agents that performed sentiment and stress analysis on text or keystroke data and several decision-level fusion analysis agents. Experiments were conducted with data from a private SNS called Pesedia [12] for discovering which analyzers were more effective at detecting user states that propagated more in the SNS. Finally, an advisor agent that generates feedback for users in a SNS was designed according to the results of the experiments, using a combination of agents that perform analyses on text and keystroke data and a set of rules.

The detection of the emotional state and stress levels of a user when he or she is writing a message are not the only sources of information available from the SNS that could be used to generate feedback. Other sources, like the historic of polarities and stress levels of users when they interacted in the past, the one of the audience where the message is about to be posted, or the topics that could be extracted from the message are examples of other sources of information that may prove to be useful at generating feedback to the user that may perform better than simple analysis on the messages at avoiding potentially negative outcomes in the social environment. In Case-Based Reasoning (CBR) systems, a reasoner remembers previous situations similar to the current one and uses them to solve the new problem [11]. The cases can contain several different features to represent a concrete situation in the system, so a CBR system could be used to combine different aspects of a user state, and also external factors to help decide what action should the system take to guide this user and potentially prevent a negative outcome.

We implemented and integrated a CBR module into a MAS, for helping it detect a case in a SNS where a user interacting could generate a negative repercussion, and for making the system able to prevent it by applying the corresponding action to each case. In this way, the system is able to take into consideration different sources of information and also previous interactions, and exploit this information for guiding users. This MAS is a variation of the one presented in [14] and the one presented in [10]. The MAS works by extracting information about a text message being posted in a SNS, which are the text of the message, the keystroke dynamics data associated, the audience that may see it, the user that posted the message, and the time of the day. The MAS agents then perform sentiment and stress analysis on text and keystroke dynamics data and use stored information about past analyses and topic detection in texts to generate more information for the CBR. Finally, the CBR module integrated into the MAS generates a case with this information, and calculates which case from its case base is the most similar to the current case to further recommend an action to be performed, such as warning the user to avoid potential future risky situations. This process is explained extensively in section 3.

In [14] we conducted a set of experiments with data from the SNS Twitter.com to determine which of the analyses used in the MAS was able to detect a state of the users that propagated the most to the replies of the messages. As a metric of propagation, the most detected state in the replies of the messages was used. In the present work, we conducted a set of experiments for discovering not only which of the analyses is able to detect a state that propagates more in the SNS, but also to compare single analyses to the prediction of the CBR module. We performed a set of experiments with people at laboratory, using Pesedia for a period of one month, and used this data to compare the analyses and the CBR module. We also performed experiments for analyzing the difference in the error of the CBR module after populating the case base with different parameters. The experiments and results are discussed in section 4.

Therefore, the contributions of this chapter are: firstly, the design and implementation of a CBR module that is able to generate cases of previous interactions of users in a SNS by using the information of the history of analyses, information retrieved from the SNS, and analyses done at the moment, and that can recommend an action to prevent potential negative repercussions in a SNS. This CBR-based approach is a way to use different information related to the user state and context of the interactions to predict potential negative outcomes in the SNS that, to the best of our knowledge, has not been performed before. Secondly, the experiments conducted with data from Pesedia for assessing what are the differences between the analyses and the CBR module in predicting a state of the user that propagates more in the SNS, so it would be more informative to generate a warning or feedback to prevent negative repercussions. Finally, we performed experiments varying parameters of the CBR module and populating the case base for discovering which set of parameters reduces the error rate of the system.

The following sections are as follows. Section 2 gives a review of state-of-art works relevant to this chapter. Section 3 describes the MAS, the CBR module integrated, and explains how the system works in general. Section 4 describes the experiments conducted with data from Pesedia, and conclusions are drawn from them. Finally, section 5 shows general conclusions and proposes future lines of work.

6.2 Related work

In the present work, we integrated a CBR module into a MAS that previously used sentiment analysis, stress analysis, and a combined version of sentiment and stress with text data and keystroke data, to help the system know what is the state of users navigating a SNS or other on-line social environments. In this way, the system is able to generate feedback for users to avoid a potential future negative situation in the social environment. For this reason, a revision of current works on CBR-based systems is performed and works where the user state is modeled and used later by a CBR in order to perform a task. Works in risk prevention and privacy aiding in SNSs are also reviewed.

CBR systems are used to generate a case out of different characteristics of the environment or user interaction, and compare this case to a database of previous cases for extracting a potential solution or action to be taken in the current situation of a system [11]. In [81], authors integrated a CBR module into a helpdesk software as a solution recommender, to be able to help operators in customer support environments. In [82], authors performed image segmentation of deformed kidneys using a CBR system and a Convolutional Neural Network (CNN) and compared them, resulting in that the CBR succeeded in performing the best image segmentation. In [83], a CBR is proposed to act as a recommendation system between users and a review site. Users are recommended by the CBR certain phrases from previous reviews found similar to the one that they are writing. The case base is populated using reviews from Amazon.com, decomposing them into words appearing on it, phrases to recommend to users, and the helpfulness rate of the review, created by Amazon.com users. Sentiment analysis using a CBR-based approach was performed in [33]. In this approach, labeled customer reviews and five different sentiment lexicons are used to populate the case base. Cases are created when a document is successfully classified by at least one lexicon. Cases contain document statistics and writing style of the review that generated it, and the solution associated which is the lexicons used to correctly predict sentiment on the review that generated the case. For prediction, the k most similar cases (1, 2, or 3 in the experiments) are retrieved, and the lexicons of the solutions are reused for the new case. In [34], domain ontology and natural language processing techniques are used to perform sentiment analysis, and case-based reasoning is used to learn from past sentiment polarizations. According to authors, the accuracy obtained by the proposed model overcomes standard statistical approaches. A CBR with a manually-constructed case base of emotion regulation scenarios for e-learning is presented in [84]. The cases represent events in which e-learners suffer from certain emotions, and the solution to the cases is the advice and phrases to regulate their emotions. Similarity between the speaker sentences and the sentences in the cases is performed to select one case to apply for emotion regulation. Authors claim that the experimental results show that the proposed method has a positive role in emotion regulation in interactive text-based applications.

There are works that apply a CBR approach to detected sentiment or opinions to perform a task with the detected user state or opinions. In [85] authors implemented a CBR system that used opinions mined from user-generated reviews to help users decide in recommendation systems; in [32] a two-layer approach was used to detect implicit customer needs in on-line reviews. The first layer uses sentiment analysis to extract explicit customer needs from reviews, using Support Vector Machines (SVM), and the second layer uses a CBR module to identify implicit characteristics of customer needs. Nevertheless, to the best of our knowledge, even when there are works that use CBR modules to improve the performance of sentiment analysis, none of the existing solutions use a CBR module in combination with different analyses of the user state (sentiment analysis, stress analysis and using different data types) or a combination of analyses and context information. Moreover, none of them use this information to generate recommendations and guide users in a system to help avoid potential future problems in on-line interaction, which may enhance the performance of the system in predicting possible negative repercussions and help avoid them.

As mentioned in previous chapters, risk prevention, user guiding, and privacy aiding in SNSs is an important topic nowadays. Privacy helping or aiding has been addressed in [65], by means of designing a user interface aiming to that purpose, with the main features of privacy in the system being visible to users by introducing privacy reminders and also customized privacy settings. In this work, privacy aiding is addressed in an indirect way, by analyzing different aspects of the mental state of the user to discover if they could be in a state that could lead to incurring risks from the interaction with other users. An example of user protection in SNSs by means of using sentiment analysis is performed in [51], since authors implemented an SNS that used adult image detection, a message classification algorithm, and sentiment analysis in text messages. The system used this information to ban users that incurred in on-line grooming and cyber-bullying. Although the use of sentiment analysis to prevent negative outcomes in SNSs has been addressed, it might prove useful to use detection of different aspects of the user mental state (e.g. sentiment analysis, stress analysis, combined analysis), together with a CBR module using context information to generate feedback that helps prevent negative outcomes, which to the best of our knowledge remains unexplored.

6.3 System description

In [14] a MAS was presented, which computed sentiment, stress, and combined analysis of sentiment and stress on text data of user messages when they interacted in a SNS. The MAS uses the SPADE multi-agent platform [62] to implement the agents of the system. There are several agents in the

MAS that perform different tasks and communicate with each other using a messaging interface based on the FIPA-ACL [72] language. Moreover, we can find three different agent types on the MAS. Firstly, there are the presentation agents, in charge of communicating with the SNS, receiving data, and sending feedback to the users. Secondly, the logic agents perform analyses on the messages and generate feedback if it is needed. Finally, the persistence agent is the one who performs the data storage and retrieval tasks. Additionally, in [10] the MAS was extended with agents that perform sentiment and stress analysis with keystroke dynamics data and a new advisor agent. This advisor agent receives the single modality analyses from the different agents and computes decision-level fusion analyses. Based on the information received and generated, and using a set of rules, it generates feedback to users by considering the output of the different analyses. The analyses used in the advisor agent were determined by performing experiments with data from Twitter.com for discovering what analyses were able to detect a state of the user that propagated more in the network.

In this work, we present a new version of the MAS, which incorporates a CBR module in the pipeline of agents. This CBR module consists of two logic type agents. One agent performs the selection of a case from the case base when a new message appears in the MAS, based on the similarity of the new case associated with this message (that is generated by this agent) to the ones in the case base, and sends the prediction of the case selected to the advisor agent for potentially generating feedback as warnings to the user when necessary. The other agent is in charge of updating the case base, adding new cases based on new messages received, and also updating the priority of cases. In this way, the agent makes the cases more likely to be selected if the predictions made with them were correct (the messages that were predicted using those cases to generate a negative or positive repercussion in the SNS did so), or more unlikely, even erasing the case when the priority is under a set threshold if the predictions made were not correct. The architecture of the system can be seen in figure 6.1.

The presentation agent has assigned the tasks of receiving data from the SNS and sending feedback to the users navigating. Regarding the case of the logic agents, we find a pipeline of agents, that perform the process needed to generate feedback to the user. This pipeline begins when the presentation agent sends the data of messages to the sentiment and stress analyzer agents

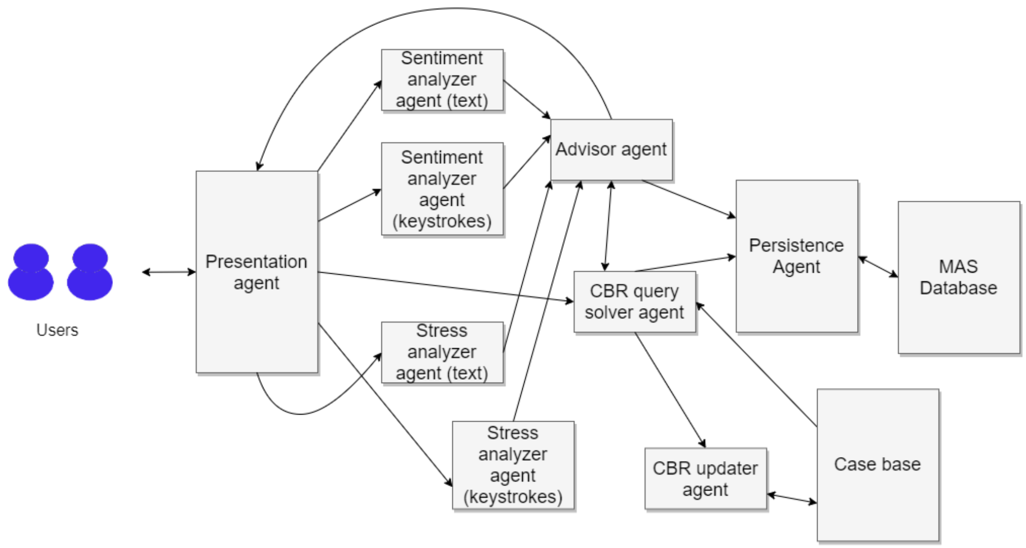


Figure 6.1: Architecture of the MAS

on text and the ones on keystroke data. When the sentiment and stress analyses have been computed, they send the outputs to the advisor agent, who sends the data gathered about the message and output of analyses to the CBR query solver agent, who is in charge of finding the best matching case on the case base. The advisor agent also sends the output of the analyses and the messages to the persistence agent to save this information in the database. The CBR query solver agent generates a case associated with the message being analyzed, performs matching with cases on the case base, and later gives the information of the solution of the selected case to the advisor agent (whether this message could generate a negative repercussion or not), for creating the feedback to the user if the message is deemed negative. The feedback is stored in the database and sent to the presentation agent, who delivers it to the SNS and to the user.

The sentiment and stress analyzer agents that work with text data perform an analysis with Artificial Neural Networks (ANN), which use text embeddings to transform the text into embedding arrays before feeding the ANNs with them. They give as output the class negative or positive sentiment in the case of the sentiment analyzer, and low and high-stress level in the case of the stress analyzer. The sentiment and stress analyzer agents working with keystroke data also use ANNs, which are fed with the arrays of keystroke features including timing and frequency of pulsation features, and give as output the same as the case of text data (positive or negative for sentiment analysis and low or high stress for stress analysis). The different analyzers using ANNs were trained with Tensorflow (`https://www.tensorflow.org`) version 1.8.0 and Keras (`https://keras.io`) version 2.2.0 in the language Python, in its version 3.5.2. The process that is carried by the CBR query solver agent and the one performed by the CBR updater agent (who updates the case base periodically) will be detailed following in this section. Finally, the persistence agent performs the actions needed to store and retrieve information about sentiment and stress labels or past predictions and messages.

6.3.1 CBR module

The CBR module, which is integrated into the MAS proposed, is formed by the CBR query solver agent and the CBR updater agent, in addition to the case base and several data structures needed for the functioning of the module. The traditional four steps in the CBR cycle are performed by these two agents. Those steps are retrieval, reuse, adaptation, and retention. The process carried by the agents for every step is elaborated in the following subsections. First of all, after receiving the output of the analyses, the author of the message, the audience of the message, and the time of the day when it was created, the CBR module creates a case with:

1. The time of the day.
2. Output of the text sentiment analyzer.
3. Output of the text stress analyzer.
4. Output of the keystroke sentiment analyzer.
5. Output of the keystroke stress analyzer.
6. Computed history of messages detected as negative or positive composed by the user writing the message sent to the CBR module, as the average from his last set of interactions.
7. Computed history of messages detected as negative or positive in messages written by the audience, as the average between all viewers (from the averaged values per user).

8. Computed topic found in the text message, using a trained LSI [86] model and the gensim[1] library.

The history for users is computed using a window of the ten past messages for each user, and the outputs of a decision-level fusion method of sentiment and stress analysis on text and keystroke dynamics data on the messages. Therefore, the history is computed as the average of the output from the decision-fusion approach detected in the last ten messages written for each user. For the case of the history of the audience of a message, an average of the history values for every user pertaining to the audience is used. The solution associated with the cases is the prediction of the module about whether the case represents messages that can create a negative repercussion in the SNS where they are posted or not. A diagram of the functioning of the CBR module, with different software agents, actors, and possible actions is shown in figure 6.2.

6.3.2 Retrieval step

The retrieval step is performed by the CBR query solver agent. In this step, cases are retrieved from the case base, and the similarity between the cases retrieved and the new case created representing the message sent to the system is computed. The CBR query solver agent is in charge of the process of generating a case from the information related to a text message. The agent also extracts cases from the case base and finds the one that does the best matching with the case generated from the message data, and store information about potential new cases to be added by the CBR updater agent.

After the case that represents the message being written in the SNS is created, the CBR query solver agent initiates a loop, checking cases in the case base, from the one with the highest priority to the lowest, but with a limitation in the number of cases that can be checked which is fixed. Nevertheless, the loop could halt if a certain number of cases matching the case generated from the message are found, which is also fixed. Those amounts were set to 100 for the maximum number of cases to be checked and 10 for the maximum number of cases to select. For assessing whether or not one case does match with the case generated from the message or not, and in

[1]https://radimrehurek.com/gensim/

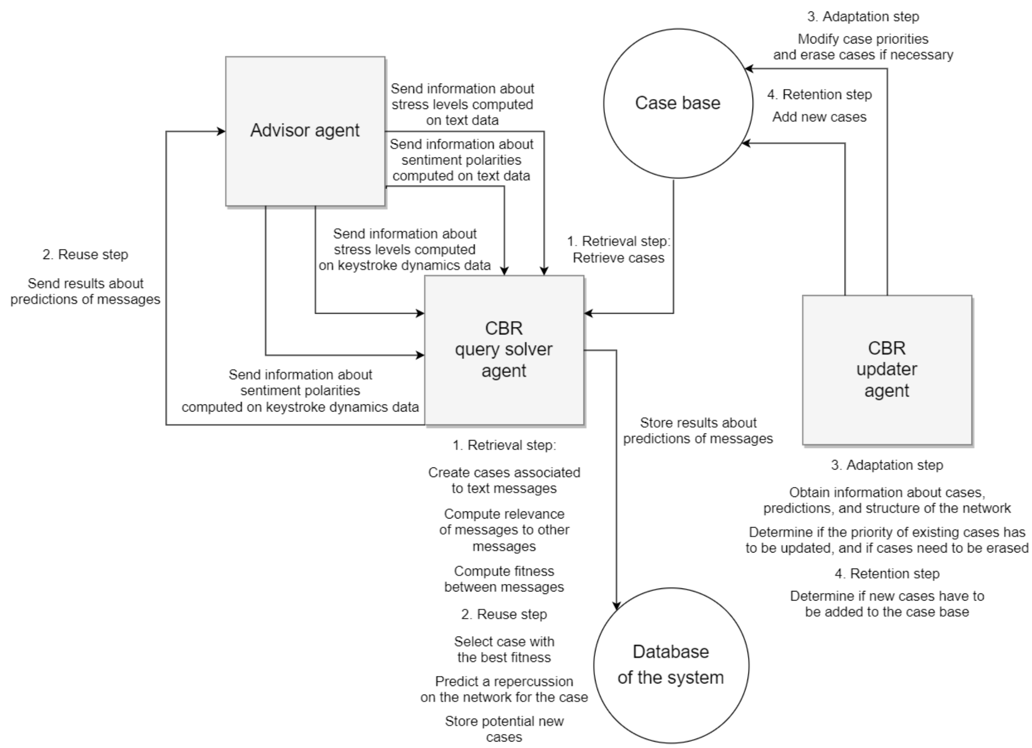


Figure 6.2: Diagram of actors and actions that form the CBR module and the elements that interact with it in the system

which grade, two functions were coded. One function checks for relevance of the case to compare (which means that the case at least has bare minimum similarities with the case from the message), and the second function computes the fitness or degree of match between the two cases, checking the similarities between each defining feature in the cases (e.g. the topics from the texts). The fitness function uses a weighted sum of the computed similarities of the features to compute the final fitness. Firstly, the relevance is checked, then if the case to compare is relevant in the sense that it has bare minimum similarities to the case generated from the message, the fitness between the two cases is computed. At the end of the process, the case that obtained the highest fitness is selected.

The relevance function process is illustrated at algorithm 1, and works as follows: run a loop for all the features that could exist in a case, and for every one check if it exists in the original message to be compared to another. If this is the case, the corresponding feature is added to a variable accounting for the features to be matched. Following, a loop runs for every feature found in the original message in the previous step. In every iteration, if the feature is found in the new message to compare to the original message, it is checked to compare the features of both cases, using comparisons according to the data type. If the comparison results in a basic match (the features are similar in at least a 50%), then 1 is added to a variable accounting for the found similarities. Finally, if the variable accounting for found similarities is equal or greater than a third of the number of features in the variable accounting for the features to be matched computed in the first loop, then the result is that the message is relevant, otherwise, it is not relevant to the original message to which it is compared. The fitness function is illustrated at algorithm 2, and works in the same way as the relevance function, except for two differences. Firstly, it checks any existing differences in the features between cases, and adds to the variable accounting for found similarities the proportion of the weight of the feature being compared equal to the degree in which the features match. Secondly, it later gives as a result of fitness the quotient between the variable accounting for found similarities and the value of a variable accounting for a sum of the weights of the features to match.

6.3.3 Reuse step

In the reuse step, the solution found in the previous step is used for the system to give an answer about whether the message sent by a user in a SNS could generate a negative repercussion in the network or social environment or not. When the CBR query solver agent has selected the case, then the information about the solution assigned to it (which is the prediction of the CBR module for if the case will generate a negative repercussion or not) is taken as the prediction for the new case associated to the message being written. This information is sent to the persistence and advisor agents. If the prediction is negative, then a warning is generated in the advisor agent and sent to the presentation agent, for delivering the feedback to the user. The persistence agent simply stores the information about the prediction in the database. The CBR query solver agent also stores new potential cases that could be later added to the case base by the CBR updater agent. For this task, the cases generated from new messages are used. If the fitness of new cases is below a threshold, meaning that the new case is different from the cases in the case base at least by the percentage given by $1-threshold$, being

```
Data: Case A and case B to compare with A
Result: Relevance of the level of similarity of B to A
sum_relevance = 0;
features_found = [];
for all the possible features of cases do
    if feature exists in A then
        Append feature to features_found;
    end
end
for each feature in features_found do
    if feature exists in B then
        Compare the value of the feature in A and B;
        if feature is integer and feature from A is equal than feature from B then
            sum_relevance+ = 1;
        else
            if feature is floating point number and absolute difference is 0.5 or less then
                sum_relevance+ = 1;
            else
                if feature is string and feature from A is 50% or more similar to feature from
                  B then
                    sum_relevance+ = 1;
                end
            end
        end
    end
end
if sum_relevance >= round(length(features_found)/3) then
    Return B similarity to A is relevant;
else
    Return B similarity to A is not relevant;
end
```

Algorithm 1: Relevance function

this threshold a value between 0 and 1, then they are added as potential new cases.

6.3.4 Adaptation step

In the adaptation step, the update of the state of the CBR module is performed to fit the ever-changing state of a real-life scenario, in which the system is supposed to be used. The process performed in this step is the adaptation of existing cases in the case base, conditioned to what the system observes and compares with its own previous predictions. For adapting the cases, priorities assigned to cases are modified. These priorities measure how well cases have performed when used for predicting and therefore if they are likely to be useful for new predictions. The CBR updater agent has a set time interval between updates, so the cases in the case base are not updated with every interaction in the SNS, but with the interactions that happened

```
Data: Case A and case B to compare with A
Result: Fitness of B with A
sum_fitness = 0;
features_found = [];
sum_weights = 0;
for all the possible features of cases do
    if feature exists in A then
        Append feature to features_found;
        sum_weights+ = weightoffeature;
    end
end
for each feature in features_found do
    if feature exists in B then
        Compare the value of the feature in A and B;
        Add to sum_fitness the fraction of the feature weight corresponding to the percentage of
          similarity;
    end
end
Return sum_fitness/sum_weights;
```

Algorithm 2: Fitness function

in that interval. The agent works in this way for potentially leading to more useful information from the repercussion of messages (one message with only one reply gives the repercussion to one message, but if there are several answers, the agent can compute the repercussion to several messages, which may be more informative for updating the priority of cases). The update of priorities is based on the computed real repercussion of messages that have been predicted to have positive or negative repercussion by the CBR query solver agent.

The CBR updater agent performs the case-update loop as is shown in algorithm 3. This agent runs a loop for all the cases that were selected by the CBR query solver agent, and for every case, another loop is done for every message that matched with it and was given a prediction based on the solution of the selected case, checking the repercussion of the message to see if it is the same than the predicted value. If it coincides, the priority of the case used for prediction is raised, if it does not coincide then the priority is decreased. At a fixed low priority limit, the cases are also erased. To check the repercussion of the messages, the MAS has data about messages and parent structure of the messages, so the CBR module agents have the information about the parents and children of messages. Finally, the repercussion is computed as the most present predicted value by the CBR module on the replies of a given message. The information about cases selected, predictions performed, and parent structure of messages is first stored by the CBR query

```
Data: Information about predictions from the CBR module and parent structure of messages
Result: Update of cases in the case base
for each case C in the case base used to predict in the previous interval between updates do
    for each prediction P done with C for a case D, performed in the previous interval between
      updates do
        Compute the most predominant predicted value for the children (replies) of the message
          associated to D;
        Compare the computed predominant value with P;
        if the predominant value coincides with P then
            Increase priority of C by 1;
        else
            Decrease priority of C by 1;
            if priority is under a set threshold then
                Delete C from the case base;
                break;
            end
        end
    end
end
Save the updated cases in the case base;
```

Algorithm 3: Update of cases in the case base

solver agent when cases are created and selected, and then used by the CBR updater agent when it is the time for updating the case base.

6.3.5 Retention step

As mentioned before, the CBR query solver agent stores information about potential new cases, which are generated when messages are sent to the MAS, and cases are created associated with them. For this task, the fitness computed of the new case with the cases on the case base are used. If the fitness of new cases is below a threshold, meaning that the new case is different from the cases in the case base at least by the percentage given by $1 - threshold$, being this threshold a value between 0 and 1, then they are added as potential new cases.

The CBR updater agent updates the case base with the messages that were assigned as potential new cases by the CBR query solver agent. If the limit of messages in the case base, which is a fixed amount, is reached, then no additional cases get added, and they remain as pending cases until the existing cases start to get erased by reaching a low priority limit in the previous step.

6.3.6 Example of the functionality of the system

When users are interacting in a SNS or other on-line social environment which is connected to the proposed MAS and publish a post in the walls of the network or in a group, before actually publishing the message, the information about the audience of the message, the user writing, the text, and the keystroke dynamics data of the message are sent to the MAS, where the presentation agent receives this data. The presentation agent sends messages to the sentiment analyzers, the stress analyzers, and the CBR query solver agent, so the analyzers can analyze the text and keystroke data and give the advisor agent the outputs of their respective analyses. The advisor agent gives the results of the analyses to the CBR query solver agent, which will use this information and the one handed out by the presentation agent to build a case representing the message being written in the SNS. The CBR query solver agent then computes the relevance and the fitness of the case created with cases in the case base and selects the best fitting case, to give a prediction of positive or negative repercussion in the network, caused by the message being written. Finally, the CBR query solver agent hands back the prediction to the advisor agent, and if it is a prediction of negative value, a warning is generated and sent to the user interacting in the SNS to prevent potential negative outcomes. Nevertheless, the user can choose to ignore the message and continue posting. A new case may get added (from the case created with the data associated with the message written in the SNS), and priorities updated in the case base if the repercussions on the SNS show that predictions made were correct or incorrect, raising or decreasing the priority of the cases, respectively. Information about predictions, sentiment polarities, stress levels, and messages are also stored in the MAS database.

6.4 Experiments with data from Pesedia

In these experiments, the main aims are two. Firstly, to investigate what are the most important features in the cases, and what is the best configuration of the CBR module for populating the case base and achieving a low error rate in the prediction of the module. Secondly, to discover if the CBR module is able to perform better than the sole use of sentiment or stress analyses (with text and keystroke data), or combined analysis methods of sentiment and stress. Therefore, in this section, the two experiments performed with data from our private SNS Pesedia will be discussed.

The dataset used is the Pesedia dataset. It contains text messages and associated keystroke dynamics data and other necessary details of the text messages for constructing the cases in the CBR module. The dataset was constructed by gathering data from the Pesedia SNS in July of 2018 and July of 2019, both times during a period of one month. Nevertheless, the number of samples gathered in 2019 is larger than the one gathered in 2018. Additionally, only those gathered in 2018 were labeled with sentiment polarity and stress level (positive or negative and low or high-stress level, respectively).

6.4.1 Experiments for assessing the best configuration of the CBR module and error rate

In this section, the experiments performed altering parameters of the CBR module before populating the case base and checking the error rate of the module will be examined. Concretely, the different configurations used and the process followed for checking the error of prediction of the module will be elaborated and results presented. We altered the following parameters of the CBR module:

- Weights of the different features in the cases. The weights associated with the case features (e.g. sentiment polarity, topic of the text) have been changed for assessing whether the system learns to predict better when giving different importance to the distinct features in the cases.
- Update interval for the CBR updater agent. The time interval between updates determines how much time we let the interactions happen in the system before the CBR module uses the information about repercussions and potential new cases to update its case base. We measure the differences in error rate when populating the case base at different update intervals.

We used the Pesedia dataset for conducting the following experimental process: a loop runs, processing an unlabeled user reply and the unlabeled original message that was replied by that message, and then a labeled reply message in every iteration. The unlabeled messages are used to feed the CBR module, so it processes them and can later update the case base according to this information. Following, the labeled message is also sent to the CBR module for processing, and the answer of the CBR is kept for computing the error of the system. Finally, a comparison is made between the label of the message and the response of the CBR module, if it is the same, then a counter of correctly analyzed messages is increased. The ratio of correctly predicted messages by the CBR is shown and stored every iteration, to be able to draw conclusions later. Messages are sent as a set of features to the CBR module (e.g. text of the message, id of the author), and the module creates a case associated by using the output of sentiment and stress analyzers, a topic model, and other information related to the messages. Therefore, there is no need to provide labels to the CBR, and both messages unlabeled and labeled can be provided to the CBR for this experiment.

For examining the error rate of the module in the process of populating the case base with different configurations while it is used for predicting, we define a metric. This metric is error of the system in a window of messages (windowed error or WE), thus the error of the system is computed for several windows of time where the system analyzes a set amount of messages. The WE_i or windowed error in the window of messages i is computed as follows:

$$WE_i = \sum_{j=c*i}^{c*i+c} e_i$$

Where e_i is 1 if the module detected a different state than the label on the labeled message at iteration j and otherwise 0, $i = 0, 1, 2, ..., n$, where n is the number of windows of messages for which we compute an error minus one, and c is a fixed constant for the size of the window, in our case we used 25 messages for each window.

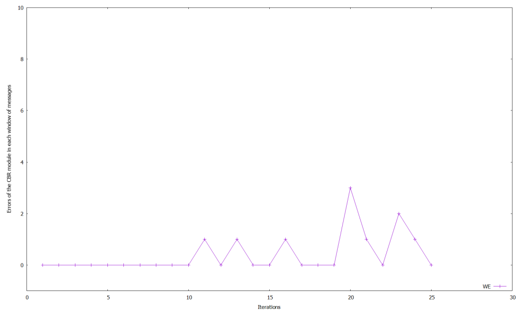
Several experiments were conducted, with different configurations of weights in the case features. Since the different possibilities of weight configurations are infinite, we created a selection of weight settings, aiming to test the configurations that led to more informative results about the performance of the CBR module. For every experiment, different update intervals for the CBR module update were tested, between 10 and 100 seconds. The configurations for each experiment are the following:

- 10_sen_str_day_and_history: Equal weight for sentiment and stress features in the cases (10), and small value in the weights of time of the day and history of the audience features (3), history of the author of the message and topic of the message weights unaltered, being 5 and 10 respectively.

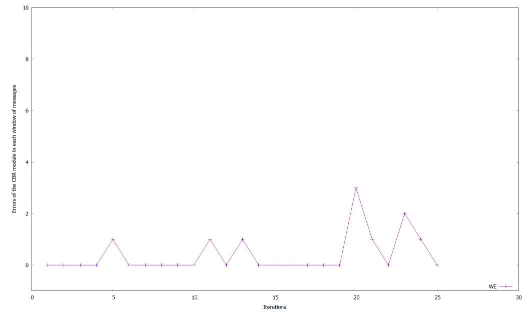
- 10_sen_str_no_day_and_history: The same case as the previous, except no value is added to the weights of time of the day and history of the audience features (0).
- 20_sen_day_and_history: Doubled value of the weight representing sentiment than the one representing stress on the message (20 and 10 respectively), and the rest of the weights unaltered with respect to the first case.
- 20_sen_no_day_and_history: The same as the previous case, except no value is added to the weights of time of the day and history of the audience features (0).
- 20_str_day_and_history: Doubled value of the weight representing stress than the one representing sentiment on the message (20 and 10 respectively), and the rest of the weights unaltered with respect to the first case.
- 20_str_no_day_and_history: The same as the previous case, except no value is added to the weights of time of the day and history of the audience features (0).

In figure 6.3 results for the experiments keeping the same weight in the sentiment and stress features while changing others are shown. Subfigures 6.3a, 6.3b, and 6.3c show the results for the 10_sen_str_day_and_history experiment with 10, 50 and 100 seconds of update interval in the CBR module, and subfigures 6.3d, 6.3e, and 6.3f show the results for the 10_sen_str_no_day_and_history experiment. In the same way, results for the experiments doubling the weight of the sentiment feature and changing others are shown in figure 6.4, and results of the experiments doubling the stress feature weight while again changing others are shown in figure 6.5.

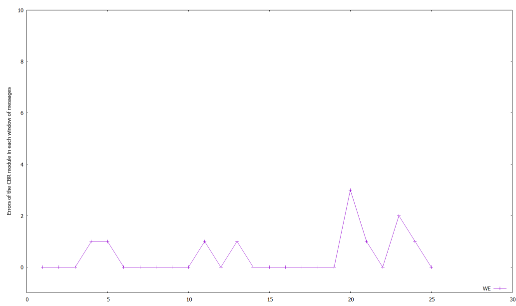
As is shown in the figures that present the error obtained by the CBR module in the 10_sen_str_day_and_history experiment, the error is low in general, observing only a small increase in the initial parts of the experiment with the largest update interval. Nevertheless, when the effect of the time of the day and history of the audience features is removed (weights set to



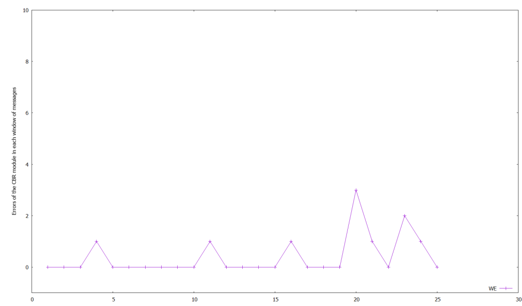
(a)



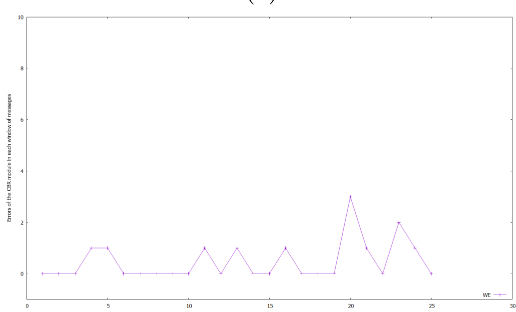
(b)



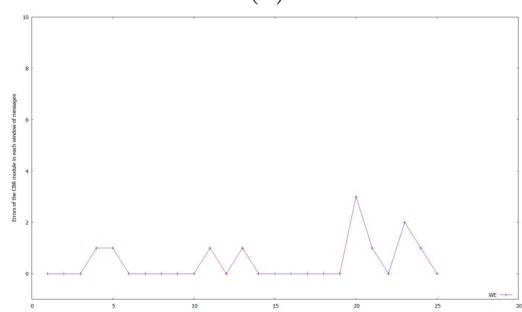
(c)



(d)



(e)



(f)

Figure 6.3: From left to right and from up to down, error in the '10_sen_str_day_and_history' experiment with 10, 50, and 100 seconds update intervals and in the '10_sen_str_no_day_and_history' experiment with the same intervals

zero) in the 10_sen_str_no_day_and_history experiment while sharing the same weights on the other features than in the previous case, it is shown still a small error in general. However, the error that in the previous experiment appeared with the largest update interval appears earlier at the 50 second

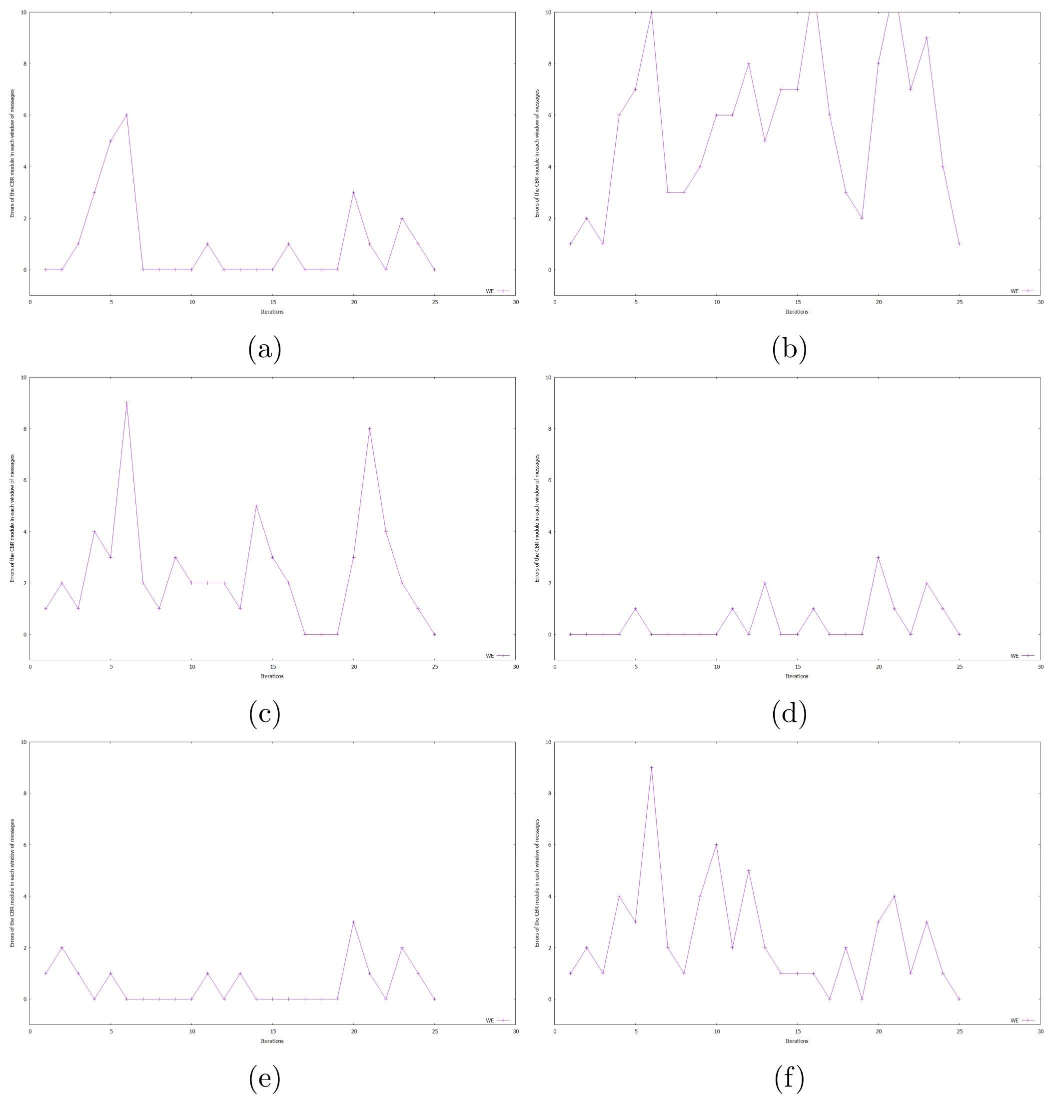


Figure 6.4: From left to right and from up to down, error in the '20_sen_day_and_history' experiment with 10, 50, and 100 seconds update intervals and in the '20_sen_no_day_and_history' experiment with the same intervals

update interval, and stays in the 100 second update interval.

Contrarily as in the previous case, in the experiments where the weights for sentiment and stress features of the cases are altered, a general trend of

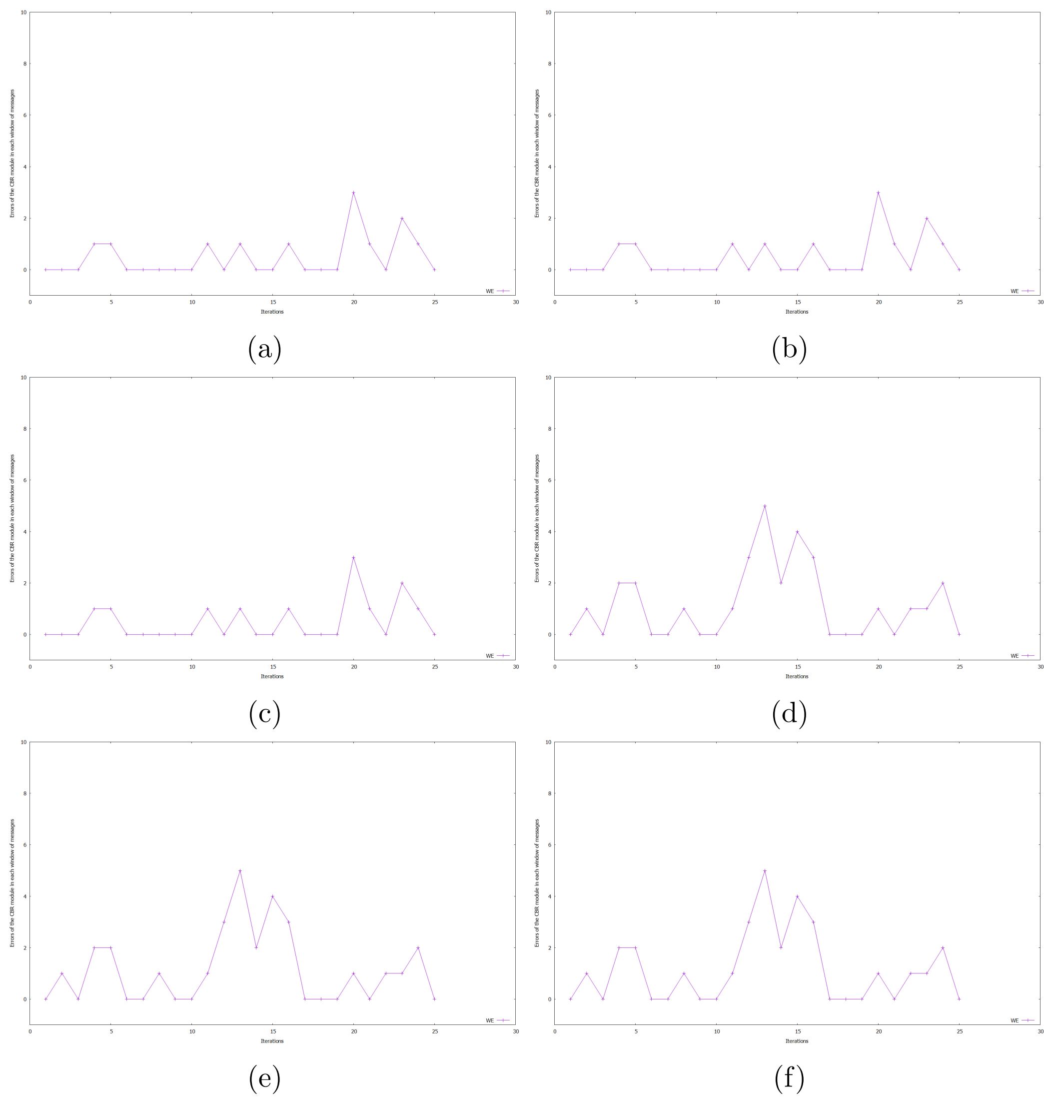


Figure 6.5: From left to right and from up to down, error in the '20_str_day_and_history' experiment with 10, 50, and 100 seconds update intervals and in the '20_str_no_day_and_history' experiment with the same intervals

more error is found in general, being higher in the case when the sentiment weight is considered two times as important than the stress one. In the case of the 20_sen_day_and_history and 20_sen_no_day_and_history experiments, the error is progressively and visibly smaller as the frequency of updates of the

CBR module is increased. Additionally, in the case of these two experiments, setting to zero the weight of the time of the day and history of the audience features appears to reduce the error to some extent.

In the case of the 20_str_day_and_history and 20_str_no_day_and_history experiments, the same effect of the time of the day and history of the audience features can be observed than on the experiments where the weights for the sentiment and stress features were the same, which is the appearance of more error in the case where these two features are set to zero. Moreover, in these two experiments, altering the update interval does not change the observed error.

To conclude, the error observed was lower when not altering the weights for the sentiment and stress features and being the same for both. Additionally, generally, the error diminishes when setting the update intervals to be more frequent (not causing it to increase in any of the experiments), demonstrating that the system is able to learn and achieve lower error. The effect of the time of the day and history of the audience of the message features showed to reduce slightly the error as a general trend, except in the experiments where the sentiment weight was considered twice as important as the stress weight, although in these experiments a high error is found, which might indicate that modifying the sentiment feature weight in this way is not ideal for reducing the error of the CBR module, as is the case when altering the weight for the stress feature in the same way.

6.4.2 Experiments for comparing the CBR module and different sentiment and stress analysis methods

We performed experiments populating the case base with Pesedia data, and then using a static version of the CBR module (without updates) for predicting negative or positive repercussion caused by messages in the Pesedia SNS, that were not the messages used for populating the case base in the first step. We also used different analyzers (individual sentiment and stress analysis and combined versions), and compare the results obtained between the CBR module and the different analyzers. The dataset used was again the Pesedia dataset.

For comparing the capacity to detect positive and negative repercussions in the SNS, we use the propagation of the detected state in the network as a metric. This metric measures the percentage of messages that were detected by an analysis method to have the same state as the ones directly influenced by it. In the case of our study, we use the replies to a message as the messages directly influenced by this message (the message that was replied). In this way, we are able to compute a metric that shows which analyzer is able to detect a state that is found more in the messages influenced by the message analyzed, therefore being able to help the system prevent potential negative outcomes (as in a negative state spreading through the network). The metric (PDV or propagation of detected value) is computed as follows:

$$PDV = \frac{messages\_with\_propagated\_state}{messages\_with\_replies}$$

Where messages_with_replies is the total amount of messages that generated replies being analyzed, and messages_with_propagated_state is the aggregated value of messages with propagated state, which are messages with the same detected state as is present in most of their replies. By the state present in most replies, we refer to the state that has the most frequency from the states detected in the replies.

Related to the analyzers, we used sentiment and stress analyzers on text data, the same with keystroke data, combined versions of sentiment and stress analyzers in text data using two versions (or and and combined analysis as a decision-level combination method of sentiment and stress analysis), and the same for keystroke data again. The or and and versions of combined analysis of sentiment and stress refer to the use of the union or intersection of the outputs of sentiment and stress analyzers, respectively, applied at decision-level (after the sentiment and stress analyses have been computed). In this way, using the or version of combined analysis, a negative class is assigned if either the sentiment polarity is negative or the stress level is high, otherwise the resulting class output is positive. For the case of the and version, a negative class is given as output when both negative sentiment and high-stress level are detected, and positive otherwise. We used our CBR module for predicting using optimized parameters. In these experiments, labels are not necessary. Therefore, the use of the labeled or unlabeled sets of samples for populating the case base and for comparing predictions is done for assessing differences between using lower or larger amounts of data when populating the case base and comparing predictions (since the sets of samples are different in size). For comparing predictions between analyzers and the CBR module, we performed experiments with the following data:

- smaller_data: Use of the labeled samples to populate the case base and the unlabeled samples to compare predictions.
- larger_data: Use of the unlabeled samples to populate the case base and the labeled samples to compare predictions.
- larger_third: Use of the first third of the unlabeled samples to populate the case base and the rest of the unlabeled samples to compare predictions.
- larger_half: Use of the first half of the unlabeled samples to populate the case base and the rest of the unlabeled samples to compare predictions.
- larger_third_and_smaller: Use of the first third of the unlabeled samples and all of the labeled samples to populate the case base and the rest of the unlabeled samples to compare predictions.
- larger_half_and_smaller: Use of the first half of the unlabeled samples and all of the labeled samples to populate the case base and the rest of the unlabeled samples to compare predictions.

In table 6.1 we show the results of PDV for different analyzers and the CBR module. For each experiment, we present two rows of results in the table. The results for analyzers working with text input are shown in the upper row, and the results for analyzers working with keystroke dynamics data are shown in the lower row. Finally, since the CBR module utilizes information from both analyses on text data and keystroke data, only one cell per experiment is presented. Every result presented in table 6.1 is the average of four different experiments, performed on the four different partitions of data in the test set of samples used. For this purpose, we partitioned the corresponding test set and used one partition for each experiment, showing the average of the four results in the table.

As shown in table 6.1, there are differences between the results obtained by the different analyzers and the CBR module. The different experiments were conducted with the aim of exploring whether the CBR module was able to obtain better results in terms of PDV than the sentiment, stress, and

Table 6.1: Comparison between analyzers and the CBR module

Experiment	***sentiment analysis***	***stress analysis***	***or combined analysis***	***and combined analysis***	***CBR module***
smaller_data	0.7055	0.9584	0.6947	0.9895	0.7395
	0.9183	0.9995	0.9179	1	
larger_data	0.6603	0.8135	0.6703	0.9413	1
	0.7638	0.9342	0.756	0.9559	
larger_third	0.6842	0.9499	0.6671	0.9899	1
	0.9173	0.9993	0.9167	1	
larger_half	0.686	0.9475	0.6666	0.9897	1
	0.9156	0.9992	0.9149	1	
larger_third_and_smaller	0.6842	0.9499	0.6671	0.9899	1
	0.9173	0.9993	0.9167	1	
larger_half_and_smaller	0.686	0.9475	0.6666	0.9897	1
	0.9156	0.9992	0.9149	1	

combined analyzers non-CBR-based (on text data and on keystroke dynamics data). An objective of performing experiments populating the case base with different data samples and different amounts of samples was performed to ascertain whether the factors of using different data or different data sizes influences or not the results. As can be seen in table 6.1, the CBR module was able to outperform the different analyzers non-CBR-based in almost every experiment, except the case of the smaller_data experiment, where the case base was populated using only the labeled data samples. In this case, when using a small number of samples, the CBR module performance is similar to the sentiment analyzer using text and the or combined analyzer of sentiment and stress on text data, but its performance is lower than the rest of the analyzers. As commented before, the results are not a consequence of using labeled samples, since labels were not used in this experiment. Nevertheless, the CBR module is able to outperform every analyzer when populating the case base with amounts of data such as the case of the larger_data experiment, and the performance did not get affected by using or mixing different partitions of data samples when populating the case base. Additionally, using only a third of the unlabeled data samples to populate the case base, and the remaining two thirds to test performance showed to be enough to not decrease the performance of the CBR module.

6.5 Conclusions and future lines of work

In this chapter, different sentiment and stress analyzers using text and keystroke dynamics data have been combined using a CBR module, and have been integrated into a MAS for guiding and recommending users that navigate on-line social platforms or environments, based on their emotional state and stress levels. The sentiment and stress analysis has been implemented in individual agents that perform sentiment and stress analysis on text data and on keystroke data. These agents communicate with other agents in a MAS for being able to receive data from messages of users in on-line social platforms in real-time, analyze the data, and hand it over to a CBR module that uses the information of the output of the analyses and context of the conversations for predicting potential negative outcomes derived from the user interaction. Since the different functions are implemented in several agents in the MAS, the tasks of the system can be parallelized to work in real-time scenarios.

The CBR approach allows for using different information about user interaction and user state together to perform predictions in on-line platforms, and recommend actions to users. In this chapter, context information such as the topic being talked about and the history of predictions of the system for the user writing and the users in the audience of a message are used together with sentiment polarity and stress level detected on text and keystroke dynamics data of messages. The CBR module implemented successfully predicts potential negative repercussions (as in negative sentiment polarity or high-stress levels spreading through users) in an on-line social platform when users write messages, using the information described above.

For assessing the performance of the proposed system, two different experiments were performed. In the first experiment, the aim was to assess the error of the system when predicting after populating the case base with different configurations of weights in the case features, and using different update intervals. For these experiments, a labeled corpus of data from a real SNS was used and compared to the prediction of the CBR module, conducting several experiments varying weights and update intervals. The experiments showed that shorter update intervals lead to less error from the CBR module, and that certain configurations in the weights of the case features lead to less error than others. For the case of the second experiment, the objective was to determine whether the CBR module was able to predict repercussions in a SNS better than individual analyzers that perform sentiment and stress analysis and analyzers that combine sentiment and stress analysis using text and keystroke dynamics data. In this experiment, several experiments were conducted changing the amount of data that was fed to the CBR module for populating the case base and for testing the performance. The CBR module managed to outperform the different analyzers except in the case of an experiment where the case base was populated using a small partition of the data from the dataset. In this case, the CBR module performance resulted similar to the sentiment analyzer using text and the or combined analyzer of sentiment and stress on text data, but lower than the other analyzers.

For future lines of work, there are unexplored possibilities. Firstly, new features could be introduced in the cases to account for additional information to the system when making predictions. One example of a feature that might prove useful is the tiredness of users, which would give more information to the system about the real-time state of users when interacting, and might lead to better performance. Machine learning techniques could be used to measure the level of tiredness in users, using text data, keystroke data, or both. In addition, a second potentially interesting line of work would be to use the system for not only predicting negative sentiment and high level of stress spread through an on-line social environment, but also use it to predict other phenomena that could be of interest to the system for guiding and recommending users, such as predicting cyber-bullying or on-line grooming, based on the detection of certain topics and states of the users interacting. Moreover, the system could be used to give different feedback to users. It could be used to warn a user that his or her message might be inappropriate if the users in the audience of the message have a recent history of negative states detected by the system, based on the output of the CBR module.

CHAPTER 7

Conclusions and future lines of work

In this thesis, we designed and implemented a Multi-Agent System (MAS) that integrated agents addressing user state detection via sentiment, stress, and combined analyses, and agents creating feedback for users navigating social on-line environments based on the prediction of the system about whether the user interaction could generate a negative repercussion or create risks in the social environment. For this reason, we propose different unimodal and multi-modal analyzers, a rule-based system for feedback generation based on the output of the analyses, and a Case-Based Reasoning (CBR) module that uses the output of the analyses, context information, and history of past predictions of the system for generating the feedback.

Initially, the MAS integrated agents performing sentiment and stress analyses on text data and a combined version of both, using a combination at decision-level, combining the output of the analyzers using the union set of negative sentiment polarity and high-stress level as negative output, and positive otherwise. Bayesian classifiers were used for the implementation of the analyzers. In the experiments performed with data from the Social Network Site (SNS) Twitter.com, it can be observed that all the analyzers implemented detected a state of the user that propagated to the replies of messages, thus leading to an analysis that could predict a negative repercussion in the SNS. This version of combined analysis worked well in domains where stress was normally found, but slightly worse than other analyzers in domains where stress was not normally present.

In a second version of the MAS, new analyzers were implemented using Artificial Neural Networks (ANNs) for performing sentiment and stress analysis. Two versions of combined analysis were used, one is the same used in the first version, and the second used the intersection set of the messages detected with negative sentiment polarity and high-stress level as the negative output label, and positive otherwise. The implementation of analyses using ANNs was performed for improving the performance of the previous classifiers used, which was achieved. The system was tested with stored data from Twitter.com and Pesedia, and also in a real-life scenario with users of Pesedia. In the experiments with stored data, it was observed that the combined analysis using the intersection performed significantly better than the other analyses at detecting states that propagated more in the network. This might be caused by the fact that the analysis receives information from two previous different analyses, and that it only gives the output as negative when both individual analyzers give as output a negative state, hence being more strict for the criteria of assigning a negative label to a message. In the experiments in a real-time scenario with Pesedia users, we created a control and test group, with the system only active in the test group. The number of messages detected with a negative state was found higher in the control group without the feedback of the system, supporting the hypothesis that the affective states of users can be useful for the system for preventing negative repercussions in an on-line social environment. Surveys were also conducted with users in the real-life experiment, for understanding their opinion about the system and the usefulness of the proposed approach. In the surveys, users indicated that they are interested in receiving alerts about potential negative outcomes from their interactions, that they think problems can indeed arise from publishing posts, and that the alerts received were not annoying. Nevertheless, despite the general trend of users to think that a problem can arise from publishing a post, in general, they answered that they do not think that their emotional state has influenced their messages.

In the third version, sentiment and stress analyzers using keystroke dynamics data from messages were created, using ANNs like in the previous case, and integrated into the MAS. Experiments with stored data from Pesedia were conducted with different versions of analyzers, using single sentiment and stress analysis on text, and on keystroke data, and using combinations at decision-level of the output of sentiment and stress analysis on one data type and on both data types. It is shown in the results that the best-performing analyses were performed by combined analyzers, one using both data types (text and keystroke dynamics) and a combination of sentiment and stress analysis, and analyzers using single data type combining sentiment and stress analysis. The thresholds for class detection in the ANNs were altered as parameters for another experiment, and making the detection more strict (higher threshold) in the ANNs that perform analysis on keystroke dynamics data and in the ANNs that perform sentiment analysis showed to benefit the best-performing analysis that combines text and keystroke data. Finally, a new version of the advisor agent, using a set of rules and the best-performing analyzers was created. It considers the different cases in which the system could be at a certain time (e.g. the tokenizer does not find a valid token in the text message, but the keystroke information can be computed, hence the combined sentiment and stress analyzer on keystroke data is used, and not the fusion combination of text and keystroke data).

In the last version of the system, a CBR module was implemented and integrated into the MAS. The module creates cases based on the output of sentiment and stress analyzers on text and the same on keystroke data, and on context information about user interaction. The module uses cases from previous interactions to predict repercussions in on-line social environments that could be caused by new interactions. Results of an experiment comparing analyzers and the CBR module showed that the module was able to detect a state of users that propagated more in the network than the one detected by any other analyzer in general. Results of experiments conducted for checking the error rate with different configurations shown that the update interval affected the error in general, decreasing it when the update interval was shorter. Additionally, certain configurations in the weights of the case features were observed to lead to less error than others.

To summarize, the proposed system fills in gaps in previous works about the detection of affective states of users with the analyzers implemented, and also implements useful feedback generation and user-guiding mechanisms. Those user-guiding mechanisms use a simple combination of the output of analyzers, a set of rules and different versions of analyzers, and finally, a

CBR module that utilizes not only the information about affective states but also about the context of user interactions and information about past interactions. Additionally, the experiments conducted with data from SNSs show the benefit of using one analyzer or other, and the CBR module or other user-guiding and feedback generation mechanism. In the end, these contributions might be able to help future developers and researchers to design and implement intelligent systems that better detect potential negative repercussions or risks from user interactions in on-line social environments, and help prevent them.

For future lines of work, there are possibilities such as investigating additional case features that could be of interest for the cases of the CBR module, and that might lead to less error. Detection of different phenomena could be also performed, such as detection of bullying topics in the text, and also different kinds of feedback for the users, such as warning the users about their history of recent states, or the history of the audiences of their messages. Moreover, alternatives to feedback and guidance, namely argumentation, could also be implemented in an attempt to improve the performance of the system for preventing the negative repercussions and potential risks, when the MAS has already detected them in an interaction. Additionally, other possibilities such as using computer vision techniques to monitor users expressions (with the user consent), could be incorporated to both improve the affective state detection and to help the system in the process of guiding (or in case that it was implemented, argumentation as well), through the use of the user visual feedback as indicators to the system about how the user feels with the system feedback.

Referring to applications of the system, future work could investigate the uses of this system in different scenarios to discover its usefulness for preventing risks and negative repercussions in on-line social platforms. Firstly, the system could be integrated as a guiding mechanism of users navigating on a real on-line platform (such as Twitter or Facebook), and the changes in interactions, emotional polarities, and stress levels monitored to determine the system efficacy. Differently, it could be integrated as a multi-platform tool to guide users navigating in different social-platforms at once, and use the information of interactions gathered in one platform to improve the efficacy of the system in another. For example, the system could use the history of states detected from users that navigate in multiple platforms to improve the effectiveness of the predictions (as in using the history of the user in one platform for predicting in another platform). In addition, other scenarios such as augmented reality applications for real-life interactions could be implemented, integrating the system in them. In this way, the system could interact with users by monitoring their voice and converting it to text, in addition to using the voice itself or other data in future modules and generating visual feedback for guiding users in real-life scenarios.

Bibliography

[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. doi:10.1007/s11423-015-9415-4.

[2] S. De Moor, M. Dock, S. Gallez, S. Lenaerts, C. Scholler, C. Vleugels, Teens and ict: Risks and opportunities. belgium: Tiro, http://www.belspo.be/belspo/fedra/proj.asp?l=en&COD=TA/00/08 (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, EU Kids Online, Deliverable D4. EU Kids Online Network, London, UK. http://eprints.lse.ac.uk/33731/ (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. doi:10.3916/C43-2014-12.

[5] J. M. George, E. Dane, Affect, emotion, and decision making, Organizational Behavior and Human Decision Processes 136 (2016) 47–55. doi:10.1016/j.obhdp.2016.06.004.

[6] M. Thelwall, Tensistrength: Stress and relaxation magnitude detection for social media texts, Information Processing & Management 53 (1) (2017) 106–121. doi:10.1016/j.ipm.2016.06.009.

[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. doi:10.1002/asi.21416.

[8] G. Aguado, V. Julián, A. García-Fornes, A. Espinosa, A review on mas-based sentiment and stress analysis user-guiding and risk-prevention systems in social network analysis, Applied Sciences 10 (19) (2020) 6746. doi:10.3390/app10196746.

[9] N. J. Shoumy, L.-M. Ang, K. P. Seng, D. M. Rahaman, T. Zia, Multimodal big data affective analytics: A comprehensive survey using text, audio, visual and physiological signals, Journal of Network and Computer Applications 149 (2020) 102447. doi:10.1016/j.jnca.2019.102447.

[10] G. Aguado, V. Julián, A. García-Fornes, A. Espinosa, Using keystroke dynamics in a multi-agent system for user guiding in online social networks, Applied Sciences 10 (11) (2020) 3754. doi:10.3390/app10113754.

[11] J. Kolodner, Case-based reasoning, Morgan Kaufmann, 2014.

[12] J. Bordera, Pesedia. red social para concienciar en privacidad, Master's thesis, Universitat Politècnica de València, Valencia, Spain (2016).

[13] 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. doi:10.3390/info9050107.

[14] G. Aguado, V. Julian, A. Garcia-Fornes, A. Espinosa, A multi-agent system for guiding users in on-line social environments, Engineering Applications of Artificial Intelligence 94 (2020) 103740. doi:10.1016/j.engappai.2020.103740.

[15] C. Zhang, D. Zeng, J. Li, F.-Y. Wang, W. Zuo, Sentiment analysis of chinese documents: From sentence to document level, Journal of the American Society for Information Science and Technology 60 (12) (2009) 2474–2487. doi:10.1002/asi.21206.

[16] M. Hu, B. Liu, Mining opinion features in customer reviews, in: Proceedings of the 19th National Conference on Artifical Intelligence, Vol. 4, no. 4, Springer, 2004, pp. 755–760.

[17] 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.

[18] B. Lu, M. Ott, C. Cardie, B. K. Tsou, Multi-aspect sentiment analysis with topic models, in: 2011 11th IEEE International Conference on Data Mining Workshops, IEEE, 2011, pp. 81–88. doi:10.1109/ICDMW.2011.125.

[19] A.-M. Popescu, O. Etzioni, Extracting product features and opinions from reviews, in: Natural language processing and text mining, Springer, 2007, pp. 9–28. doi:10.3115/1220575.1220618.

[20] 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. doi:10.1145/945645.945658.

[21] 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.

[22] D. Borth, R. Ji, T. Chen, T. Breuel, S.-F. Chang, Large-scale visual sentiment ontology and detectors using adjective noun pairs, in: Proceedings of the 21st ACM international conference on Multimedia, ACM, 2013, pp. 223–232. doi:10.1145/2502081.2502282.

[23] Q. You, J. Luo, H. Jin, J. Yang, Robust image sentiment analysis using progressively trained and domain transferred deep networks., in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI Press, 2015, pp. 381–388.

[24] L. Kaushik, A. Sangwan, J. H. Hansen, Sentiment extraction from natural audio streams, in: 2013 IEEE International Conference on

Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 8485–8489. doi:10.1109/ICASSP.2013.6639321.

[25] S. Deb, S. Dandapat, Emotion classification using segmentation of vowel-like and non-vowel-like regions, in: IEEE Transactions on Affective Computing, Vol. 10, no. 3, IEEE, 2019, pp. 360–373. doi:10.1109/TAFFC.2017.2730187.

[26] J. Deng, Z. Zhang, E. Marchi, B. Schuller, Sparse autoencoder-based feature transfer learning for speech emotion recognition, in: 2013 humaine association conference on affective computing and intelligent interaction, IEEE, 2013, pp. 511–516. doi:10.1109/ACII.2013.90.

[27] F. Mairesse, J. Polifroni, G. Di Fabbrizio, Can prosody inform sentiment analysis? experiments on short spoken reviews, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2012, pp. 5093–5096. doi:10.1109/ICASSP.2012.6289066.

[28] I. Kanluan, M. Grimm, K. Kroschel, Audio-visual emotion recognition using an emotion space concept, in: 2008 16th European Signal Processing Conference, IEEE, 2008, pp. 1–5.

[29] J. Chen, B. Hu, L. Xu, P. Moore, Y. Su, Feature-level fusion of multimodal physiological signals for emotion recognition, in: 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2015, pp. 395–399. doi:10.1109/BIBM.2015.7359713.

[30] M. A. Nicolaou, H. Gunes, M. Pantic, Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space, IEEE Transactions on Affective Computing 2 (2) (2011) 92–105. doi:10.1109/T-AFFC.2011.9.

[31] M. S. Hossain, G. Muhammad, M. F. Alhamid, B. Song, K. Al-Mutib, Audio-visual emotion recognition using big data towards 5g, Mobile Networks and Applications 21 (5) (2016) 753–763. doi:10.1007/s11036-016-0685-9.

[32] F. Zhou, R. Jianxin Jiao, J. S. Linsey, Latent customer needs elicitation by use case analogical reasoning from sentiment analysis of online product reviews, Journal of Mechanical Design 137 (7). doi:10.1115/1.4030159.

[33] B. Ohana, S. J. Delany, B. Tierney, A case-based approach to cross domain sentiment classification, in: International Conference on Case-Based Reasoning, Springer, 2012, pp. 284–296.

[34] F. Ceci, A. L. Goncalves, R. Weber, A model for sentiment analysis based on ontology and cases, IEEE Latin America Transactions 14 (11) (2016) 4560–4566. doi:10.1109/TLA.2016.7795829.

[35] L. M. Vizer, L. Zhou, A. Sears, Automated stress detection using keystroke and linguistic features: An exploratory study, International Journal of Human-Computer Studies 67 (10) (2009) 870–886. doi:10.1016/j.ijhcs.2009.07.005.

[36] R. Feldman, Techniques and applications for sentiment analysis, Communications of the ACM 56 (4) (2013) 82–89. doi:10.1145/2436256.2436274.

[37] K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Transactions on Knowledge and Data Engineering 28 (3) (2016) 813–830. doi:10.1109/TKDE.2015.2485209.

[38] R. Ji, D. Cao, Y. Zhou, F. Chen, Survey of visual sentiment prediction for social media analysis, Frontiers of Computer Science 10 (4) (2016) 602–611. doi:10.1007/s11704-016-5453-2.

[39] L. Li, D. Cao, S. Li, R. Ji, Sentiment analysis of chinese micro-blog based on multi-modal correlation model, in: 2015 IEEE International Conference on Image Processing (ICIP), IEEE, 2015, pp. 4798–4802. doi:10.1109/ICIP.2015.7351718.

[40] S. D. Gunawardhane, P. M. De Silva, D. S. Kulathunga, S. M. Arunatileka, Non invasive human stress detection using key stroke dynamics and pattern variations, in: 2013 International Conference on Advances in ICT for Emerging Regions (ICTer), IEEE, 2013, pp. 240–247. doi:10.1109/ICTer.2013.6761185.

[41] P.-M. Lee, W.-H. Tsui, T.-C. Hsiao, The influence of emotion on keyboard typing: an experimental study using auditory stimuli, PloS one 10 (6) (2015) e0129056. doi:10.1371/journal.pone.0129056.

[42] M. M. Bradley, P. J. Lang, The international affective digitized sounds (2nd edition; iads-2): Affective ratings of sounds and instruction manual, University of Florida, Gainesville, FL, Tech. Rep. B-3.

[43] 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. `doi:10.4018/978-1-5225-5191-1.ch088`.

[44] D. Rosaci, Cilios: Connectionist inductive learning and inter-ontology similarities for recommending information agents, Information systems 32 (6) (2007) 793–825. `doi:10.1016/j.is.2006.06.003`.

[45] 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. `doi:10.1109/MIS.2015.98`.

[46] 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. `doi:10.1016/j.elerap.2013.07.003`.

[47] R. Cissée, S. Albayrak, An agent-based approach for privacy-preserving recommender systems, in: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems, Association for Computing Machinery, 2007, pp. 1–8. `doi:10.1145/1329125.1329345`.

[48] A. Singh, A. Sharma, Maicbr: A multi-agent intelligent content-based recommendation system, in: Information and Communication Technology for Sustainable Development, Springer, 2018, pp. 399–411. `doi:10.1007/978-981-10-3920-1_41`.

[49] C. Villavicencio, S. Schiaffino, J. A. Diaz-Pace, A. Monteserin, Y. Demazeau, C. Adam, A mas approach for group recommendation based on negotiation techniques, in: International Conference on Practical Applications of Agents and Multi-Agent Systems, Springer, 2016, pp. 219–231. `doi:10.1007/978-3-319-39324-719`.

[50] J. Rincon, F. de la Prieta, D. Zanardini, V. Julian, C. Carrascosa, Influencing over people with a social emotional model, Neurocomputing 231 (C) (2017) 47–54. `doi:10.1016/j.neucom.2016.03.107`.

[51] A. Upadhyay, A. Chaudhari, S. Ghale, S. Pawar, et al., Detection and prevention measures for cyberbullying and online grooming, in: 2017 International Conference on Inventive Systems and Control (ICISC), IEEE, 2017, pp. 1–4. doi:10.1109/ICISC.2017.8068605.

[52] M. Camara, O. Bonham-Carter, J. Jumadinova, A multi-agent system with reinforcement learning agents for biomedical text mining, in: Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics, Association for Computing Machinery, 2015, pp. 634–643. doi:10.1145/2808719.2812596.

[53] M. Almashraee, D. M. Diaz, R. Unland, Sentiment classification of on-line products based on machine learning techniques and multi-agent systems technologies., in: Industrial Conference on Data Mining-Workshops, IBaI Publishing, 2012, pp. 128–136.

[54] G. Lombardo, P. Fornacciari, M. Mordonini, M. Tomaiuolo, A. Poggi, A multi-agent architecture for data analysis, Future Internet 11 (2) (2019) 49. doi:10.3390/fi11020049.

[55] F. Schweitzer, D. Garcia, An agent-based model of collective emotions in online communities, The European Physical Journal B 77 (4) (2010) 533–545. doi:10.1140/epjb/e2010-00292-1.

[56] A. El Fazziki, F. Z. Ennaji, A. Sadiq, D. Benslimane, M. Sadgal, A multi-agent based social crm framework for extracting and analysing opinions, Journal of Engineering Science and Technology 12 (8) (2017) 2154–2174. doi:10.1109/AICCSA.2016.7945700.

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

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

[59] 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, Asian Federation of Natural Language Processing, 2013, pp. 1107–1111.

[60] Y. Seroussi, I. Zukerman, F. Bohnert, Collaborative inference of sentiments from texts, in: Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization, Springer-Verlag, 2010, p. 195–206. doi:10.1007/978-3-642-13470-8_19.

[61] H. Nakamura, S. Mise, T. Mine, Personalized recommendation for public transportation using user context, in: 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), IEEE, 2016, pp. 224–229. doi:10.1109/IIAI-AAI.2016.204.

[62] 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. doi:10.1145/1160633.1160866.

[63] J. Villena Román, S. Lana Serrano, E. Martínez Cámara, J. C. González Cristóbal, Tass - workshop on sentiment analysis at sepln, Procesamiento del Lenguaje Natural 50 (2013) 37–44.

[64] J. Villena-Román, J. García-Morera, S. Lana-Serrano, J. C. González-Cristóbal, Tass 2013 - a second step in reputation analysis in spanish, Procesamiento del Lenguaje Natural 52 (2014) 37–44.

[65] W. Xie, C. Kang, See you, see me: Teenagers' self-disclosure and regret of posting on social network site, Computers in Human Behavior 52 (C) (2015) 398–407. doi:10.1016/j.chb.2015.05.059.

[66] 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. doi:10.1016/j.im.2006.03.001.

[67] 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.

[68] 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. doi:10.1016/j.eswa.2008.01.070.

[69] 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. doi:10.1016/j.future.2019.09.016.

[70] 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. doi:10.1016/j.future.2018.07.021.

[71] 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. doi:10.1109/ACCESS.2018.2886314.

[72] P. D. O'Brien, R. C. Nicol, Fipa-towards a standard for software agents, BT Technology Journal 16 (3) (1998) 51–59. doi:10.1023/A:1009621729979.

[73] 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

[74] A. Mehrabian, Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament, Current Psychology 14 (4) (1996) 261–292. doi:10.1007/BF02686918.

[75] C. Epp, M. Lippold, R. L. Mandryk, Identifying emotional states using keystroke dynamics, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2011, pp. 715–724. doi:10.1145/1978942.1979046.

[76] A. Kołakowska, Towards detecting programmers' stress on the basis of keystroke dynamics, in: 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), IEEE, 2016, pp. 1621–1626.

[77] F. Huang, X. Zhang, Z. Zhao, J. Xu, Z. Li, Image–text sentiment analysis via deep multimodal attentive fusion, Knowledge-Based Systems 167 (2019) 26–37. doi:10.1016/j.knosys.2019.01.019.

[78] S. Poria, I. Chaturvedi, E. Cambria, A. Hussain, Convolutional mkl based multimodal emotion recognition and sentiment analysis, in: 2016 IEEE 16th international conference on data mining (ICDM), IEEE, 2016, pp. 439–448. doi:10.1109/ICDM.2016.0055.

[79] S.-H. Lau, Stress detection for keystroke dynamics, Ph.D. thesis, Carnegie Mellon University (May 2018). doi:10.1184/R1/6723227.v3.

[80] M. Ulinskas, R. Damaševičius, R. Maskeliūnas, M. Woźniak, Recognition of human daytime fatigue using keystroke data, Procedia computer science 130 (2018) 947–952. doi:10.1016/j.procs.2018.04.094.

[81] S. Heras, J. Á. García-Pardo, R. Ramos-Garijo, A. Palomares, V. Botti, M. Rebollo, V. Julián, Multi-domain case-based module for customer support, Expert Systems with Applications 36 (3) (2009) 6866–6873. doi:10.1016/j.eswa.2008.08.003.

[82] F. Marie, L. Corbat, Y. Chaussy, T. Delavelle, J. Henriet, J.-C. Lapayre, Segmentation of deformed kidneys and nephroblastoma using case-based reasoning and convolutional neural network, Expert Systems with Applications 127 (2019) 282–294. doi:10.1016/j.eswa.2019.03.010.

[83] D. Bridge, P. Healy, The ghostwriter-2.0 case-based reasoning system for making content suggestions to the authors of product reviews, Knowledge-Based Systems 29 (2012) 93–103. doi:10.1016/j.knosys.2011.06.024.

[84] F. Tian, P. Gao, L. Li, W. Zhang, H. Liang, Y. Qian, R. Zhao, Recognizing and regulating e-learners' emotions based on interactive chinese texts in e-learning systems, Knowledge-Based Systems 55 (2014) 148–164. doi:10.1016/j.knosys.2013.10.019.

[85] K. Muhammad, A. Lawlor, R. Rafter, B. Smyth, Great explanations: Opinionated explanations for recommendations, in: International Conference on Case-Based Reasoning, Springer, 2015, pp. 244–258. doi:10.1007/978-3-319-24586-7_17.

[86] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American society for information science 41 (6) (1990) 391–407. doi:10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.