MixedEmotions

Social Semantic Emotion Analysis for Innovative Multilingual Big Data Analytics Markets

D4.10 Social Context Analysis for Emotion Recognition, Final Version

<table>
<thead>
<tr>
<th>Project ref. no</th>
<th>H2020 644632</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project acronym</td>
<td>MixedEmotions</td>
</tr>
<tr>
<td>Start date of project (dur.)</td>
<td>01 April 2015 (24 Months)</td>
</tr>
<tr>
<td>Document due Date</td>
<td>30 September 2016</td>
</tr>
<tr>
<td>Responsible for deliverable</td>
<td>UPM</td>
</tr>
<tr>
<td>Reply to</td>
<td>J. Fernando Sánchez-Rada</td>
</tr>
<tr>
<td>Document status</td>
<td>Final</td>
</tr>
<tr>
<td><strong>Project reference no.</strong></td>
<td>H2020 644632</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td><strong>Project working name</strong></td>
<td>MixedEmotions</td>
</tr>
<tr>
<td><strong>Project full name</strong></td>
<td>Social Semantic Emotion Analysis for Innovative Multilingual Big Data Analytics Markets</td>
</tr>
<tr>
<td><strong>Document name</strong></td>
<td>MixedEmotions_D4.10_29_09_16_SocialContextAnalysisForEmotionRecognition_UPM</td>
</tr>
<tr>
<td><strong>Security (distribution level)</strong></td>
<td>PU</td>
</tr>
<tr>
<td><strong>Contractual delivery date</strong></td>
<td>30/09/2016</td>
</tr>
<tr>
<td><strong>Deliverable number</strong></td>
<td>D4.10</td>
</tr>
<tr>
<td><strong>Deliverable name</strong></td>
<td>Social Context Analysis for Emotion Recognition</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>Report</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>Final</td>
</tr>
<tr>
<td><strong>WP / Task responsible</strong></td>
<td>Universidad Politécnica de Madrid (UPM)</td>
</tr>
<tr>
<td><strong>Contributors</strong></td>
<td>J. Fernando Sánchez-Rada, Álvaro Carrera, Óscar Araque, Carlos A. Iglesias</td>
</tr>
<tr>
<td><strong>EC Project Officer</strong></td>
<td>Martina Eydner</td>
</tr>
</tbody>
</table>
# Table of Contents

EXECUTIVE SUMMARY ........................................................................................................................... 4

1 INTRODUCTION ....................................................................................................................................... 5

2 SOCIAL CONTEXT MODEL AND METRICS ............................................................................................. 6
  2.1 INTRODUCTION ..................................................................................................................................... 6
  2.2 GENERAL MODEL ................................................................................................................................... 6
  2.3 TWITTER API AND ATTRIBUTES ........................................................................................................ 7
  2.4 SOCIAL METRICS FOR SOCIAL CONTEXT ANALYSIS ........................................................................ 9
    2.4.1 Social Metrics Types ...................................................................................................................... 9
    2.4.2 Social Metrics provided in MixedEmotions .................................................................................. 10

3 SOCIAL MEDIA CRAWLER AND SOCIAL ANALYSIS ........................................................................... 15
  3.1 INTRODUCTION .................................................................................................................................... 15
  3.2 SOCIAL CONTEXT DATA ....................................................................................................................... 16
  3.3 INFLUENCE METRICS .......................................................................................................................... 17
    3.3.1 Tweet influence ............................................................................................................................. 17
    3.3.2 User influence ............................................................................................................................... 17

4 TEMPORAL ANALYSIS FOR SOCIAL CONTENT .................................................................................... 18
  4.1 INTRODUCTION ..................................................................................................................................... 18
  4.2 SPREADING IN BRAND REPUTATION MANAGEMENT ......................................................................... 18
  4.3 NEWS SPREAD EXPERIMENT ............................................................................................................ 23

5 EMOTION RECOGNITION BASED ON SOCIAL CONTEXT ...................................................................... 25
  5.1 INTRODUCTION ..................................................................................................................................... 25
  5.2 RELATED WORK ..................................................................................................................................... 25
  5.3 COMMUNITY DETECTION ALGORITHMS ............................................................................................. 26
  5.4 USER AND COMMUNITY EMOTION CLASSIFICATION ...................................................................... 27
    5.4.1 Weighted average .......................................................................................................................... 28
    5.4.2 Learning methods ........................................................................................................................... 29
    5.4.3 Experimentation and results ........................................................................................................ 30

6 CONCLUSIONS AND FUTURE WORK .................................................................................................. 31

7 REFERENCES ........................................................................................................................................... 32
Executive Summary

This document D4.10 describes the result of the task T4.5 “Social Context Analysis for Emotion Recognition” and integrates the previous version of this document D4.9. Sentiment and Emotion Recognition in text frequently take as an input only the textual content of the interaction (e.g. tweet). Nevertheless, there is a social context that can add valuable information for improving sentiment and emotion recognition. This deliverable defines a social context model, which is composed of two main entities: user and content. Users are social media users that author content, and are characterised by their profile, behaviour and social network. Content is social media data authored by users and it’s characterised by its metadata and associated social activity (reshares, favourites, etc.). This deliverable details the metrics that have been defined in the social context model, a tool that calculates those metrics, as well as the performed experiments to evaluate the impact of the social features on emotion recognition.
1 Introduction

Information shared in social networks is not isolated. The meaning of a particular piece of content (e.g. a Tweet, a Facebook status or a blog post) may only be fully understood when its social context is taken into consideration. In fact, social context has an effect on the behaviour of users in social networks. The behavioural changes due to perceived relationship with other people, organizations and society in general are known as social influence [Sun & Tang 2011]. The strength of social influence depends on many factors such as the strength of relationships between people in the networks, the users of the network or temporal effects. Many commercial applications are based on the social influence phenomenon [Sun & Tang, 2011], such as marketing, advertising or expert recommendation. Some applications, like viral marketing, aims at influence maximisation.

In contemporary research and industrial practice, most emotion and sentiment analysis systems try to detect collective emotion of social media content through the analysis of user or content attributes. Nevertheless, they usually do not consider how social influence affects users emotions and sentiments. In existing literature, several researchers have studied how social influence can complement sentiment and emotion analysis. West et al. showed that the assumption of homophily in networks can improve polarity detection in short texts [West et al. 2014]. One interesting aspect of the mix of emotion recognition and social network analysis is that the inherent complexity of emotions could also enable us to create richer models of social behaviour. For instance, by mapping traditional categories to a Valence, Arousal and Dominance model [Guerini et al. 2015] we are able to motivate a theory to explain why users in social news have a predilection for certain types of content.

In the MixedEmotions project, we are interested in the analysis and characterization of the social influence phenomenon for its application in several use cases. First, the detection of the most relevant shared content (e.g. tweets or posts) and users (e.g. influencers) provides a path for micro-analysis of opinions in brand monitoring and content recommendation scenarios. Second, emotion propagation patterns can be used for both analysis and prediction of expected social influence of a message. Finally, social features can improve sentiment analysis and emotion techniques. This can be specially relevant in microblogging based social networks such as Twitter, where the short length of the content makes the task very complex.

The remainder of the document is structured as follows. First, Section 2 presents a social context model to characterise a social graph and its features. Section 3 presents the architecture of a prototype system that has been developed to provide the social context from the Twitter network. Section 4 details the temporal analysis that describes the detection of sentiment propagation patterns in the network. Based on that social context, Section 5 shows the experimentation results for emotion recognition integrating social context information. Finally, we conclude and describe plans for future work in Section 6.
2 Social Context Model and Metrics

2.1 Introduction
Social Networks are graphs whose nodes are people and the edges are some kind of relationship between individuals, e.g. a network of cellphone users and their calls. This deliverable will focus on users of online social networks and their interactions, using Twitter as a working example. All the concepts presented can be applied to any other microblogging network, and other social media platforms such as to Facebook, although some restrictions may apply. However, other examples tend to suffer of lack of widespread adoption or of a more restrictive interfaces and policies. The remainder of this section shows the Social Context General Model (Section 2.2), the Twitter API and its attributes used as data source (Section 2.3) and the social metrics defined in the module for social context analysis (Section 2.4).

2.2 General model
The proposed Social Context Model for MixedEmotions, shown in Figure 1, consists in two main entities: Users and Social Media Content. Users belong to a social network and share social media. Social Media Content is the content (e.g. tweets, messages, comments, etc.) shared by users in their interactions in the social network. Two other entities are relevant for our analysis: Topic and Community. Social Media Content can be annotated or classified with one or more Topics (e.g. keywords, hashtags, etc.). In the same way, people’s conversations can be classified into topics, and we can deduce users’ interests based on these topics. Based on the analysis of the relationships between users, several types of communities can be analysed [Darmon et al. 2014]: structure-based (who are the friends and followers), activity-based (whose activity influences your activity), topic-based (what do you talk about) and interaction-based (whom do you communicate with). In our model, we analyse all these types of communities: structure-based communities (user networks), topic-based (topic networks), activity-based (metrics of retweets, popularity, tweet relevance and user relevance). Moreover, we have defined a new type of community: emotion communities (who feels like you). Homophily leads to connections in social networks and includes an emotion component [Fan et al, 2014]. Several studies [Fan et al, 2014, Kramer et al, 2014, Tan et al. 2011] analyse emotion contagion or propagation in social networks, based on how the emotional state is propagated in the social network via contagion and social influence mechanisms. Thus, we consider the analysis of the emotions of users, identify communities of users with the same emotion, and provide these features for improving sentiment and emotion recognition in social contents.

Given the pilot requirements of MixedEmotions, the Social Context Model has been designed with specific reference to the Twitter social media platform, but could be extended to other social networks (e.g. Facebook, YouTube, etc.) easily. In fact, given the lack of emotion annotated corpus for Twitter, we have used a dataset of the Chinese social network Weibo. In conjunction with the preparation of this deliverable, MixedEmotions produced an emotion annotated Twitter corpus (described in deliverable 4.4), however it was not available in time for inclusion in the analyses presented here.

D4.10 Social Context Analysis for Emotion Recognition
Page 6 of 37
As a result of the analysis of the social interactions, the Social Context Model provides a number of metrics for users, social media content (tweets) and the emotions of users and communities, as detailed in the next section.

![Social Context Model Diagram]

**Figure 1: Main entities of the Social Context Model**

### 2.3 Twitter API and attributes

The information needed to compute social metrics relies on information provided by the Twitter API\(^1\). Twitter provides two APIs (Application Programming Interfaces) for retrieving information: the REST API and the Streaming API.

The REST API provides a programmatic model to read and write Twitter data. The REST API has a rate limit of the order of 15 requests in each 15 minutes window. The Streaming API provides a programmatic model to access Twitter’s global stream of Tweet data. It keeps a persistent HTTP connection opened. For a random sample of the total stream of Tweets, the API has a limit of 1%. Both APIs are commonly combined to leverage their functionalities and limits.

The API is structured as a set of objects. Tweet objects are the basic atomic building block. Tweets can be embedded, replied to, liked, unliked and deleted. Tweets are posted (tweeted) by Users from a certain Place or Coordinates. In addition, Tweets can include a set of Entities (media, url, hashtags, etc.). Users can post, follow, create lists, be mentioned, have a home timeline or look up content. Users can also declare one or more contributors which are users that post on behalf on the user. A detailed view of the relationships and attributes is provided in Figure 2.

\(^1\)Documentation for Twitter API: https://dev.twitter.com/overview/documentation
The attributes provided by Twitter for Users and Tweets are shown in Tables 1 and 2, respectively.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>followers_count, favourites_count, friends_count, statuses_count, listed_count, follow_request_sent, following, contributors_enabled, notifications, verified, protected</td>
</tr>
<tr>
<td>id_str</td>
<td>profile_background_color, profile_background_title, profile_background_url, profile_background_image_url, profile_use_background_image, profile_banner_url, profile_sidebar_fill_color, profile_sidebar_border_color, profile_image_url, profile_link_color, profile_text_color, profile_image_url, default_profile_image, default_profile</td>
</tr>
<tr>
<td>name</td>
<td></td>
</tr>
<tr>
<td>url</td>
<td></td>
</tr>
<tr>
<td>screen_name</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td></td>
</tr>
<tr>
<td>created_at</td>
<td></td>
</tr>
<tr>
<td>is_translator</td>
<td></td>
</tr>
<tr>
<td>location</td>
<td></td>
</tr>
<tr>
<td>time_zone</td>
<td></td>
</tr>
<tr>
<td>geo_enabled</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 2: Class model of Twitter API [VisualParadigm, 2015]**

The attributes provided by Twitter for Users and Tweets are shown in Tables 1 and 2, respectively.
Table 2: Main Tweet attributes provided by Twitter API

2.4 Social metrics for social context analysis

In this section, we follow a quantitative approach to analyse Twitter data by providing a set of metrics that characterise the social context of tweets. First, we review the state of the art to summarise the social metrics types in Section 2.4.1 and then describe the metrics provided in the MixedEmotions platform in Section 2.4.2.

2.4.1 Social Metrics Types

Twitter has received great attention from the research community. Several works have addressed the definition of metric frameworks, focused on communication patterns [Bruns et al. 2014, Brunes et al., 2013] and influence metrics [Riquelme et al. 2016], which are usually combined. Several researchers have extended the Twitter metric system defined by Pal [Pal et al., 2005]. This Twitter metric system defines a set of metrics to include the frequency of tweets, replies, retweets, mentions and graphs.

Based on these studies, twitter metrics can be classified into user metrics, group metrics, influence metrics content metrics and temporal metrics, as shown in Figure 3.

![Social Metrics Taxonomy](image)

**Figure 3: Social Metrics Taxonomy**

**User metrics** describe the activity of users and their impact on other users. A common approach [Bruns et al, 2013, Riquelme et al, 2016] is to organise them into activity metrics, popularity metrics and structural metrics. **Activity metrics** describe metrics associated to user posts (e.g. number of sent messages (tweets, replies or retweets), statistics about the entities (media, URLs,
hashtags, etc.) of these tweets, and other aspects such as whether the user edited the replies they made. **Popularity metrics** (so called *visibility metrics* [Bruns et al. 2013]) refer to the effect of the sent messages on the rest of users (e.g. number of mentions, favourites or retweets received). **Structural metrics** refer to the social relationships of the user (e.g. number of followers, number following, betweenness, etc.). **Content metrics** refer to the metrics associated to a specific content (e.g. a tweet). Usually these metrics are associated to its popularity: number of retweets, replies, etc.

While user metrics can help to identify more active and popular users, it is frequently useful to aggregate users, since user activity in most social networks tends to follow a power law, that is, there is only a few active users and a long tail of passive users. A common way to define groupings [Bruns et al. 2014] is the rule of 10/90 (10% of most active users, 90% of remaining users) or 1/9/90 (1% of most active users, 9% of most active users not included in the 1%, and remaining 90%) [Tedjamula, 2005]. **Group metrics** consider metrics in aggregated users. The previously defined metrics (activity and popularity) can be defined for every identified group. For example, tweets sent by every group.

The metrics above analyse the behaviour of users in the full dataset. Nevertheless, it can be interesting to analyse a specific period of time and the temporal dynamics of the interactions. **Temporal metrics** analyse the communication patterns over time. The number of unique users participating in a period of time or network growth per user in a period of time are examples of temporal metrics [Beres et al, 2016].

**Influence metrics** [Riquelme et al, 2016; Noro et al. 2016] attempt to identify the most influential users or contents. Usually influential users are associated to hub nodes, whose influence can be spread through their network. These metrics are very diverse and usually elaborate the previously defined metrics. Riquelme et al. [Riquelme et al, 2016] classifies influence metrics into general metrics based on social network analysis (closeness, betweenness, etc.), those based on metrics on PageRank, topical sensitive and predictive metrics.

### 2.4.2 Social Metrics provided in MixedEmotions

We have selected the social metrics listed in Table 3, which are provided through a REST API detailed in MixedEmotions deliverable D5.4.

For the MixedEmotions platform, we have selected the metrics used by Noro et al. [Noro et al., 2016], which are topic sensitive influence metrics, since we are interested in determining the influence of users and contents about a particular topic (i.e. a brand) in the brand monitoring pilot. The main reason of the selection of this framework is that it provides a formulation with allows the calculation of tweets and user impact on real time, since it is based on two phases. The preparation phase creates a user-tweet graph representing the relation between users and tweets based on the past user activities related to the topic, calculates the influence of each user and tweet in the topic and then defines two types of each user’s power, called “Voice” and “Impact”, indicating “how much voice the user has on the topic” and “how much impact the user has on the other users’ tweets on the topic”. In the second phase, called the main phase, the relevance of newly-arrived tweets to the topic is calculated according to the Voice and the Impact score of the
users who posted, retweeted, or replied to each of the tweets, then the tweets are ranked by the relevance score. The two phases are processed independently. Once the preparation phase is completed, the main phase can return the final result any time.

<table>
<thead>
<tr>
<th>Object</th>
<th>Metrics type</th>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tweet</td>
<td>Influence</td>
<td>relevance</td>
<td>Tweet Relevance Score: shows the relevance of a tweet based on the “voice” of the original user</td>
</tr>
<tr>
<td>Content</td>
<td></td>
<td>retweetCount</td>
<td>Number of Retweets this Tweet has</td>
</tr>
<tr>
<td></td>
<td></td>
<td>favouriteCount</td>
<td>Number of Favourite marks this Tweet has</td>
</tr>
<tr>
<td>Structural</td>
<td></td>
<td>followers</td>
<td>Number of users following this user</td>
</tr>
<tr>
<td></td>
<td></td>
<td>following</td>
<td>Number of users this user is following</td>
</tr>
<tr>
<td></td>
<td></td>
<td>followRatio</td>
<td>Ratio followers/following</td>
</tr>
<tr>
<td></td>
<td></td>
<td>followRelation Score</td>
<td>Follow Relation Score: Measures the quality of the relations between this users and other users of the network</td>
</tr>
<tr>
<td>Popularity</td>
<td></td>
<td>hIndexFav</td>
<td>h-index calculated from the number of Favourite marks of the last 100 Tweets of the user. <a href="https://en.wikipedia.org/wiki/H-index">https://en.wikipedia.org/wiki/H-index</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>hIndexRt</td>
<td>h-index calculated from the number of Retweets of the last 100 Tweets of the user. <a href="https://en.wikipedia.org/wiki/H-index">https://en.wikipedia.org/wiki/H-index</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>repliedRatio</td>
<td>Ratio of user's tweets that receive a reply</td>
</tr>
<tr>
<td>Activity</td>
<td></td>
<td>replyRatio</td>
<td>Ratio of user's tweets that are a reply to other users</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tweetRatio</td>
<td>Measures the proportion of the tweets published by the user which are related to the topic</td>
</tr>
<tr>
<td>Influence</td>
<td></td>
<td>influence</td>
<td>User Influence Score: measures the “amount of attention” that a user receives from other users</td>
</tr>
<tr>
<td>[Noro et al., 2016]</td>
<td></td>
<td>openinfluence</td>
<td>User Influence calculated using the OpenInfluence algorithm: <a href="https://en.paradigmadigital.com/dev/openinfluence/">https://en.paradigmadigital.com/dev/openinfluence/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>relevance</td>
<td>User Relevance Score: This metric is the combined score of the Tweet Rate Score, the User Influence Score and the Follow Relation Factor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>voice</td>
<td>User Voice: measures the ability of a user to post or retweet influential tweets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>impact</td>
<td>User Impact: measures the ability of a user to improve the relevance of a tweet depending on their influence</td>
</tr>
</tbody>
</table>

Table 3: Social Metrics provided in MixedEmotions
For sake of completeness, we detail below how influence metrics are calculated based on Noro et al. [Noro et al, 2016].
• **Tweet Rate Score**: This metric measures the proportion of tweets related to the topic that a user posts or retweets. We use the following formula to calculate it:

$$TR(u) = \frac{|\{t \in T | t.user.id = u.id\}|}{|\text{Total}(u)|}$$

Where $t$ is a tweet posted by the user that is relevant to the topic and $\text{Total}(u)$ are the total amount of tweets posted by the user during the topic search duration.

• **Follow Relation Factor**: This metric shows how well a user is interconnected with the rest of users. In order to calculate it, we create a relation matrix, to build an adjacency matrix. Then we multiply this matrix to obtain the score. The formulas used are the following:

$$A_f(u_i, u_j) = \begin{cases} 
1 & \text{if } u_i \text{ follows } u_j \\
0 & \text{otherwise}
\end{cases}$$

$$B_f(u_i, u_j) = \begin{cases} 
\frac{A_f(u_i, u_j)}{\sum_k A_f(u_i, u_k)} (1-d) + \frac{d}{|U|} & \text{if } \sum_k A_f(u_i, u_k) \neq 0 \\
\frac{1}{|U|} & \text{otherwise}
\end{cases}$$

$$f = B_f^T f$$

Where $|U|$ is the total number of users, $d$ is a damping factor, and $f$ is a column vector of the FR score. At the end, the score is normalised:

$$FR(u_i) = \frac{f(i)}{\max_k f(k)}$$

• **User Influence Score**: This metric measures “how much attention” a user gets from other users.

$$u = B_t^T t$$

Where $u$ is a column vector of the UI score. $B_t$ is the tweet-to-user relation matrix, formed by tweets posted/retweeted and authors. We create three matrices with the relationships between the users and the tweet:

$$A_t(t_i, u_j) = \begin{cases} 
1 & \text{if } t_i \text{ is posted/retweeted by } u_j \\
0 & \text{otherwise}
\end{cases}$$

$$A_r(u_j, t_i) = \begin{cases} 
1 & \text{if } u_j \text{ retweets/replies to } t_i \\
0 & \text{otherwise}
\end{cases}$$

$$A_s(u_j, t_i) = \begin{cases} 
1 & \text{if } u_j \text{ follows at least 1 user who post/retweets } t_i \\
0 & \text{otherwise}
\end{cases}$$

We transform them in order to get two relationship matrices:
Now, in order to get the scores, we do the following:

\[ B_t(t_i, u_j) = \frac{A_t(t_i, u_j)}{\sum_k A_t(t_i, u_k)} \]

\[ B_a(u_j, t_i) = \begin{cases} \frac{A_r(u_j, t_i)}{\sum_k A_r(u_j, t_k)} (1 - d) + \frac{A_s(u_j, t_i)}{\sum_k A_s(u_j, t_k)} d \quad \text{if} \quad \sum_k A_r(u_j, t_k) \neq 0 \\ \text{otherwise} \end{cases} \]

The process is repeated until \( k = 10000 \)

Lastly, the user influence score is normalised.

\[ u_0 = \left( \frac{1}{|U|}, \frac{1}{|U|}, \ldots, \frac{1}{|U|} \right) \quad t_0 = \left( \frac{1}{|T|}, \frac{1}{|T|}, \ldots, \frac{1}{|T|} \right) \]

\[ k = 1 \]

\[ t_k = B_a^T u_{k-1} \quad u_k = B_t^T t_k \]

\[ k = k + 1 \]

The process is repeated until \( k = 10000 \)

Lastly, the user influence score is normalised.

\[ UI(u_j) = \frac{u(j)}{\max_k u(k)} \]

- **User Relevance Score**: This metric is the combined score of the Tweet Rate Score, the User Influence Score and the Follow Relation Factor. We use the previously calculated metrics to obtain the score.

\[ UserRel(u) = TR(u)^{wr} \times UI(u)^{wi} \times FR(u)^{wj} \]

Where \( w_r, w_i, \) and \( w_j \) are weights for the different metrics.

- **Tweet Influence Score**: This metric measures “how much attention” a tweet receives from the users. This metric is calculated along side with the User Influence Score.

\[ t = B_a^T u \]

Where \( t \) is the column vector of the Twitter Influence Score. \( B_a \) is the user-to-tweet relation matrix, consisting of tweets and the users that paid attention to them.

- **User Impact**: This metric measures the ability of a user to improve the relevance of a tweet depending on their influence. To calculate it we use the following formula:

\[ Impact(u) = \begin{cases} \frac{UI(u)}{|Relate(u)| - \sigma_i} \times \left( 1 - d \right) + \frac{UI(u)}{|T|} \times d \quad \text{if} \quad |Relate(u)| > 0 \\ \frac{UI(u)}{|T|} \quad \text{otherwise} \end{cases} \]

Where \( UI(u) \) is the user influence of user \( u \), \( Relate(u) \) is the set of tweets that user \( u \) has retweeted or replied to, and \( T \) is the set of all tweets in the search.
• **User Voice**: This metric measures the ability of a user’s posts and reposts to be influential.

\[
Voice_t(u) = \frac{1}{|Tweet(u)| + \sum_{t \in Tweet(u)} TI(t)}
\]

\[
Voice_r(u) = \frac{1}{|Retweet(u)| + \sum_{t \in Retweet(u)} TI(t)}
\]

Where Tweet(u) is the set of all tweets posted by user u and Retweet(u) is the set of all tweets that the user u has retweeted. \( \sigma \) is a smoothing factor.

• **Tweet Relevance Score**: This metric shows the relevance of a tweet based on the “voice” of the original user and the impact of the users that have posted, retweeted or replied to this tweet. We use the following formula:

\[
TweetRel(t) = \alpha \times VR(t) + (1 - \alpha) \times IR(t)
\]

\[
VR(t) = Voice(t, user)
\]

\[
IR(t) = \sum_{u \in Related(t)} Impact(u)
\]

Where \( \alpha \) is a weight factor.

• **OpenInfluence**: This metric calculates the relevance of users based on their popularity (number of followers and mentions received) and their influence, which is calculated using the number of tweets a user posts per week, the number of retweets received per week and the number of users that receive or watch the content produced.

\[
r = \log_2(f + m_w) + \log_2( (f \times t_w) + (\bar{a}r \times r_w) )
\]

Where \( f \) is the number of followers of the user, \( m_w \) is the number of mentions per week, \( t_w \) is the number of tweets posted per week, \( ar \) is the retweet audience and \( r_w \) are the retweets per week.
3 Social Media Crawler and Social Analysis

3.1 Introduction
The previously defined metrics can be calculated by software developed during the project, called Scaner, which is provided with an open source license and available at Github: http://github.com/mixedemotions/scaner. The experiments exposed in this document have been executed and evaluated using this tool. Scaner provides social context analysis as a service through an REST API, making it accessible from the web. To meet that goal, the Scaner module has been developed to meet the following key requirements:

- **Periodical Analysis**: Social Network are very active sites. The content is ephemeral, and relevance change in time. Also, one of the features of our system is temporal evolution analysis of the content or a user. This means that our system must be constantly updating the data and acquiring more information, in order to do periodical analyses.

- **Modular**: The system should be modular in order to provide flexibility and further improvements or extended functionalities. It also helps to customise the system to the needs of a particular application.

- **Topic centric**: As said before, context is the key to understand the influence of a user or his content. Relevance is dependent on context factors, such as the subject, time, scope, etc. This means that our system must make independent analyses of the same data, one for each topic or category. Different topic analyses will offer different results for the same content.

- **Periodical Data Updates**: Our system needs to be continuously connecting to Twitter in order to extract new data or update existing data.

- **Accessible by Web**: Being a system based on Social Network Data Analysis, our users will demand access via the web, so a web interface and API must be provided.

- **Offline processing**: The system should be able to process the data and deliver the results on demand when the calculation process is time consuming or can not be done in real time.

---

2The details of REST API can be found in D5.4 [MixedEmotions2016]
Taking into account these criteria, Scaner has been designed following a modular architecture, as shown in Figure 4. The system is composed of different modules with differentiated functions. In order to address big data scalability, a task manager has been defined so that long tasks can be processed and managed. A web server provides access to the system though a REST API. Several tasks have been defined which are handled by different modules: scraping and crawling both tweets related to a topic and the social network associated to a user or a tweet; processing social metrics, and emotion annotation of users, communities and tweets. The results of the tasks are stored in the OrientDB graph oriented database.

The remainder of this section shows the results of applying social context extraction and calculating metrics using the Scanner module on sample data.

### 3.2 Social Context Data

In this section, we expose a brief overview of the data collected with the Crawler included in the Social Context Analysis module for testing this module, as shown in Table 4.

<table>
<thead>
<tr>
<th>Topic</th>
<th>#Users</th>
<th>#Tweets</th>
<th>#Following</th>
<th>#Follower</th>
<th>#Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>All topics</td>
<td>3,504,483</td>
<td>1,564</td>
<td>420,242,857</td>
<td>636,623,486</td>
<td>6,655,758</td>
</tr>
<tr>
<td>Kutxa</td>
<td>15,575</td>
<td>32</td>
<td>15,208,110</td>
<td>19,668,798</td>
<td>2,620</td>
</tr>
<tr>
<td>BBVA</td>
<td>2,803,521</td>
<td>791</td>
<td>635,317,326</td>
<td>607,019,301</td>
<td>965,483</td>
</tr>
</tbody>
</table>
Table 4: Overview of data collected with Social Crawler. The default category includes tweets not included in other categories.

To obtain this data, the Crawler used Twitter API to collect context information, such as users network connections, including friends and followers for each user. The final number of users in the network (including the friends and followers of each tweeting user), starting with the authors of one and a half thousand tweets was three and a half millions of users with their own followings and followers.

3.3 Influence metrics

This section shows some interesting results from the global analysis performed using Scancer. Section 3.3.1 shows interesting examples of the most influential tweets and Section 3.3.2 examples of the most influential users detected from the three and a half millions stored in our database.

3.3.1 Tweet influence

Tweet influence is normalised to a range between 0 and 1. Thus, our most influential tweet is labelled with 1; the least influential one, with 0.00057; and the averaged influence value is 0.09, as shown in Table 5.

<table>
<thead>
<tr>
<th>#Influence max</th>
<th>#Influence min</th>
<th>#Influence avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00057715301</td>
<td>0.09007276425285718</td>
</tr>
</tbody>
</table>

Table 5: Global Values of Tweet Influence metrics

The text of one of the most influential tweet is “Leroy Merlin invierte 22 millones para abrir una tienda en A Coruña https://t.co/Cs6NFS2BSm #LaVozdeGalicia” (Leroy Merlin invests 22 million euros to open a new store in A Coruña)tweeted by CorunaNoticias.

In contrast, the least influential tweet text is “RT @LigaEndesa: As ha logrado el Laboral Kutxa @Baskonia terminar con la imbatibilidad del @valenciabasket en la #LigaEndesa https://t.co/en” (RT @LigaEndesa: Laboral Kutza managed to break @valenciabasket unbeaten record in #LigaEndesa) tweeted by Gorka007.

3.3.2 User influence

This section shows some user metrics and the user with highest value on each of them, as exposed in Table 6.
### Table 6: Maximum Values for User Influence Metrics

<table>
<thead>
<tr>
<th>Value</th>
<th>0.21994220862</th>
<th>0.00033414766</th>
<th>2.46863741669</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>AnaMiranda1997</td>
<td>palvalos</td>
<td>Lindalotte86</td>
<td>palvalos</td>
<td>Mr_MakeItLukEZ</td>
</tr>
</tbody>
</table>

4 Temporal Analysis for Social Content

#### 4.1 Introduction

The study of variation over time in social networks is of special interest for three reasons. First of all, because content is typically ephemeral, and it gets user attention only for a short span of time. Hence, the importance of content is directly proportional to how recent it is. Second of all, it is important due to phenomena that are linked to the way in which content spreads. For instance, the cascades of reshares, which some works aim to predict [Cheng et al., 2014]. Other works have modelled emotions in social networks as infectious diseases [Hill et al. 2010], where the order of interactions and exposure time are key. Third of all, because content can be aggregated on different time scales, giving relevant information such as movement in popularity [Jansen et al, 2009].

In this document, we present two experiments in temporal analysis of content, described in detail in the following sections. The first one analyses a sample of the data used in the Brand Reputation scenario, Pilot 2 of MixedEmotions project. The second one is focused on the Social News scenario, where we study the role of emotions in propagation.

#### 4.2 Spreading in Brand Reputation Management

In order to collect a more realistic and complete dataset to perform social context analysis, a brand monitoring dataset for pilot 2 has been crawled from twitter. Given five brands, we first collect a sample set of tweets in which each tweet contains some descriptions of one of the given brands, over the course of a week. Then, each tweet is tagged with the entities it refers to and annotated with emotion and sentiment, using the modules in the platform. Important keywords are also identified from each tweet. Furthermore, to collect a more complete social graph, those sample tweets were loaded into the Scaner module that extracted their user profiles and activities, including their latest tweets and retweets. The tweets and retweets were further cleaned by removing those that did not have enough retweets, and those where the capture was incomplete. The results are summarised in the following Table 7.
<table>
<thead>
<tr>
<th>Entity</th>
<th># of related (re)tweets</th>
</tr>
</thead>
<tbody>
<tr>
<td>kutxabank</td>
<td>2037</td>
</tr>
<tr>
<td>INGdirect</td>
<td>1651</td>
</tr>
<tr>
<td>ing direct</td>
<td>1651</td>
</tr>
<tr>
<td>Kutxabank</td>
<td>2037</td>
</tr>
<tr>
<td>infocaixa</td>
<td>9112</td>
</tr>
<tr>
<td>bbva_esp</td>
<td>44262</td>
</tr>
<tr>
<td>lacaixa</td>
<td>9112</td>
</tr>
<tr>
<td>BBVA</td>
<td>44262</td>
</tr>
<tr>
<td>banco bilbao vizcaya</td>
<td>44262</td>
</tr>
<tr>
<td>la caixa</td>
<td>9112</td>
</tr>
<tr>
<td>caixabank</td>
<td>9112</td>
</tr>
<tr>
<td>Laboral Kutxa</td>
<td>633</td>
</tr>
<tr>
<td>laboralkutxa</td>
<td>633</td>
</tr>
<tr>
<td>bbv argentaria</td>
<td>44262</td>
</tr>
<tr>
<td>Caixabank</td>
<td>9112</td>
</tr>
<tr>
<td>ingdirectes</td>
<td>1651</td>
</tr>
<tr>
<td>ing-direct</td>
<td>1651</td>
</tr>
<tr>
<td>laboral-kutxa</td>
<td>633</td>
</tr>
<tr>
<td>cuenta verde de microbank</td>
<td>9112</td>
</tr>
</tbody>
</table>

*Table 7: Entities and quantity of tweets for Pilot 2 Data*

In this dataset, we performed two different analyses. First, we took a look at the pattern of retweets of every relevant tweet in the dataset. Unsurprisingly, given the distribution of the dataset, most relevant tweets correspond to one of the entities related to the BBVA bank. These results are shown in Figure 5. As we can see, the highlighted tweets are distributed across the whole observation period. A more interesting visualization of this data is provided in Figure 6, where we use the time of the original tweet as reference. This illustration shows the similarities in propagation of most of the relevant tweets in the dataset. Furthermore, Figure 7 shows these values, normalised to the maximum retweet count value. The plain similarities, save for various outliers, hint at common patterns of diffusion, which have been studied in previous works [Yang et al, 2011].
Figure 5: Evolution of the number of retweets over several days. Each color represents a different original tweet.

Figure 6: Evolution of retweets, relative to the reception of the original tweet
After the preliminary analysis of the propagation of the main tweets in our dataset, we analysed in greater detail the propagation patterns of the most popular tweets in our dataset. To avoid having long queues due to the very few last retweets, we excluded from the analysis the last 2% of retweets. In particular, we want to highlight the differences between the two most retweeted tweets in the dataset (Figures 8 and 9, respectively). As can be seen in the illustrations, both tweets reach a similar number of retweets, but the trend is very different. Whereas the second tweet rises quickly and monotonously, which is the pattern followed by most tweets in the dataset, the first tweet has several peaks of retweets during its lifespan. These peaks seem to coincide with rush hours in Twitter, and they are likely due to the longer lifespan of the Tweet in question. In fact, the two visible peaks in the second tweet’s retweet curve match two of the peaks in the first one’s, at about 6PM and 9PM.
Figure 8: Retweets of the most popular tweet in the dataset

Figure 9: Retweets of the second most retweeted tweet in the dataset
4.3 News Spread Experiment

The second experiment aims to measure the impact of emotions in the spread of news on social media. We set out to answer the question “does the emotion of a Tweet affect how it propagates?”. It is motivated by recent studies that show that negative and positive information spreads differently [Park et al 2012], and others that have unveiled distinctive patterns of propagation in online media [Yang et al. 2011].

The experiment focuses on social news. Social News is a particular type of social media that is limited to news and events. This content is usually posted by accounts that are associated with a traditional or online news media, and it is linked to an extended version of the article. For instance, the Twitter account of Deutsche Welle (@dwnews) tweets about every new post on their website.

To collect the dataset, we compiled a list of influential social news accounts. The final list includes Twitter accounts such as The Guardian (“@guardian”) and The New York Times (@nytimes). We used the Streaming API to track their activity between 17 March until 20 April 2015. This activity includes tweets, retweets and mentions, all of which were processed for the analysis. We performed sentiment and emotion analysis of the most popular tweets (as measured by the sum of retweets and favourite count).


<table>
<thead>
<tr>
<th># of users</th>
<th>926034</th>
</tr>
</thead>
<tbody>
<tr>
<td># of original Tweets</td>
<td>329505</td>
</tr>
<tr>
<td># of retweets (captured)</td>
<td>2176809</td>
</tr>
<tr>
<td># of favourites</td>
<td>4967687</td>
</tr>
<tr>
<td>Average #Tweets/Account followed</td>
<td>8237</td>
</tr>
</tbody>
</table>

Table 8: Joined Data for Social News Dataset

The core of the experiment consists on comparing the temporal evolution of retweets and favourite count of the selected social news tweets. Thus, the first part was to calculate basic parameters of that temporal evolution to compare different tweets. For every tweet, we got the maximum number of retweets, maximum number of favourites, and the time (since the creation of the tweet) that it took to reach 25, 50, 75 and 90% of those maximums.
In this experiment, we found that we have been able to detect sentiment propagation patterns but emotion propagation evolution has not been possible with the available emotion detection models. This may be due to the fact that the results of the emotion analysis are very poor, which can be observed by manual inspection of the data. This hints that the corpora available for emotion annotation in this domain are not enough to get an analysis subtle enough to detect emotions from social news, and new resources would be needed in order to find patterns for different emotions. Nevertheless, when using polarity and subjectivity detection services, we were able to notice notable differences between positive and negative content, both in audience reached and temporal evolution. On average, it took negative tweets half the time to reach twice the retweet count of the positive ones. This confirms previous research [Rozin et al 2014] and the results obtained in similar studies on Facebook [Kramer et al 2014]. A visual representation of the evolution of the most popular tweets can be seen in the Figures 10 and 11.

Additionally to these tweets, we have gathered follower/friend relationships. Given the size of the network, we selected those users that retweeted 100 or more times, which results in a list of the 925 top retweeters. From those users, we collected their followers, a total of 7,730,067 follower-followed links. This is the network of users that were potentially exposed to the original retweets. The resulting data is currently being used to test the Social Analysis Module with real world data.
Figure 11: Evolution of Retweets in the Social News dataset, normalised and with a common origin

5 Emotion Recognition based on Social Context

5.1 Introduction
With the growth of user-generated content in social networks [Liu, 2015], interest in opinion mining has increased. In this context, the analysis typically relies solely on the text of the messages (e.g., in Twitter, the text of the tweets). Nevertheless, in the environment of social networks there is a huge quantity of information that is not related to a specific message, but to the users who create and read it. We refer to this type of information as the social context of the user [Cambria et al., 2013]. Social context consists not only of user characteristics (e.g., description, number of friends, etc.), but also the connections of the user to other users, or even their interactions. Also, as described in [Cambria et al., 2013], the social context will continue to grow in importance, as intelligent systems will be able to access more information from a vast network of users. In light of this, in this section we present several social-based approaches to emotion detection on a public social dataset. These methods exploit social information of each user, with the goal of predicting the emotion of each user.

5.2 Related work
Social relations between users and related meta-data to them can be exploited in order to gain insight into the sentiment and emotion analysis problems. The approach presented by [Deitrick et al., 2013] uses community detection and community partition algorithms to enable a more robust sentiment analysis. They achieved this improvement by combining both sentiment and
community features. Another work where sentiment analysis improves through the use of the social context is presented in [Ren et al., 2016], where conversations from the user are extracted. These features are combined with the information from the textual representation, incorporating all this information into a single model.

While sentiment analysis can benefit from social data, it is possible to improve community detection algorithms by incorporating sentiment-based information. As presented in [Lam et al. 2016], sentiment-based communities detection gains from performing contextual sentiment analysis, taking into account many forms of information about the user, such as network structure and conversational and topic contexts. In a similar work, [Stieglitz et al., 2013] use the user’s sharing information to predict if a tweet is going to be retweeted or not. They found that more emotionally charged messages tend to be more shared than the neutral ones.

Moreover, [Jenders et al., 2013] has shown that social-related features can give insights into the task of predicting which tweets are going to have a high influence. That is, they use many features, including social ones, to predict with high accuracy if a tweet is to become viral. Another work where the context of social media is effectively used is presented in [Becker et al., 2010], where user-provided annotations are combined with automatically created information, both including textual and non-textual information.

5.3 Community detection algorithms

Just like there are many different ways to define and construct a network, there are plenty of ways to define a community. Each of these definitions will result in one or more algorithms to calculate such communities, with their own set of characteristics and shortcomings. For a complete review and comparison of different algorithms, please see [Orman et al, 2011] and [Papadopoulos et al, 2010].

In our experiments, we have mainly used three algorithms to generate communities: a modularity optimization hierarchical algorithm (Louvain method [Blondel et al, 2008]), an edge-betweenness optimization hierarchical algorithm and one based on random walks (walktrap [Pons et al, 2005]).

The advantage of the hierarchical algorithms is that they allow us to select different levels of granularity in the final communities. To detect communities, these algorithms follow a divisive hierarchical method. The process works can be applied to both edges and nodes. At each step, a metric such as edge betweenness (number of shortest paths between two nodes that include the edge) is calculated for every edge. The edge with highest metric is removed, and the values of are recalculated. As more edges are removed, the initial single component of the network starts breaking down into smaller separate clusters, until all edges are removed and all nodes are isolated. The end result is a dendrogram as seen in Figure 12. The downside of these algorithms is that they are rather slow (typically $O(mn(m+n))$, or $O(n^3)$ on sparse graphs).
To remedy the slowness of these algorithms, we decided to make use of the faster walktrap algorithm. The algorithm aims to find densely connected subgraphs via random walks. A random walk is a path in the network that is formed iteratively by randomly choosing an edge from the last node in the path, adding it and its corresponding end node to the path.

We have also introduced what we call “Emotion Communities”. Instead of treating every kind of interaction equally, we have also opted to generate a separate network for each of the emotion labels in the original dataset. Reducing the number of interactions for each network also results in sparser networks. To compensate for this, the threshold, or minimum number of interactions between two users to consider the presence of an edge, has to be reduced as well.

Lastly, since there are many possible definitions of a community, it is not trivial to compare the result of two different community detection algorithms that optimise different metrics. Visual inspection of the resulting graph might be an option for smaller networks, but for a larger networks a more systematic approach is needed. Community detection studies typically employ the normalised mutual information (NMI). Given two communities, C1 and C2, their normalised mutual information is defined as:

\[
\text{NMI}(C_1, C_2) = 1 - \frac{1}{2} \left( \frac{H[X|Y]}{H[X]} + \frac{H[Y|X]}{H[Y]} \right)
\]
5.4 User and Community Emotion classification

The objective of this analysis is to predict users’ emotions based on the social information contained in the described datasets. To this end, we develop a user emotion-aware model that describes interactions between users in a social network (e.g., Weibo) in terms of their emotional features. Once this model has been defined, the next step consists on applying data mining techniques to estimate the emotion of a user.

The proposed model for users is as follows. Each user can be represented as a $d$ dimensional vector of emotions $e$, where each component represents the proportion of emotion the user shows. This vector of emotions is obtained computing the normalised distribution of emotions showed by the user$^3$. That is, the emotion vector $e$ is normalised. Alternatively, if considering a single emotion for a user, the $\text{argmax}$ operation is applied to the vector, so that the most frequent emotion showed by the user is the one considered.

Continuing with this model, we consider that a user is contained in a network, connected to $n_u$ users through $n_u$ edges that have associated a weight $w$, and where each user has a set of emotions $e$. The parameters $w$ represent the frequency of the communication between users. In the Figure 13, a graphical representation of this model is shown, where a user with unknown emotion is connected to 4 other users whose predominant emotion is represented by different colours.

---

$^3$This vector fulfills $\sum_i e_i = 1$
Applying this model to the previously described datasets, two different techniques can be used for emotion prediction. The first one consists of taking a weighted average of the emotion data for each neighbouring user, as the prediction for that user’s emotions. The second one makes use of a learning algorithm trained for the emotion prediction task. The second technique can be divided in two types of predictions: (i) prediction of a single emotion, when considering a user emotion as a unique label; and (ii) prediction of an emotion vector, considering the user emotion as an ensemble of several emotions.

5.4.1 Weighted average

Inspired by the insights provided by [Fan et al., 2014], we derive an expression that summarises the emotions on the users connected analysed one, where the users considered are the ones directly connected. This follows the intuitions given in [Fan et al., 2014], where the greatest correlation was found between directly connected users. This formula takes a weighted average of the emotions of the $n_v$ users connected to user $k$ using the edges’ weights, and normalises in order to obtain a normalised emotional vector. The formula can be written as:

$$e_k = \frac{1}{\sum_i w_i} \left( w_1 e_1 + w_2 e_2 + \ldots + w_{n_v} e_{n_v} \right)$$
Defining the vector of edge weights $w$ as $w = [w_1, w_2, \cdots, w_n]^T$ and the matrix $E \in \mathbb{R}^{n_u \times d}$ with entries $E_{i,j}$ as the value for emotion $j$ of neighbouring user $i$, the predicted vector $e_k$ can be expressed as:

$$e_k = \frac{E^T w}{1^T w}$$

This metric can be thought of as an estimation of the influence of each neighbouring users’ emotion on the considered user, filtered (or weighted) with the frequency of their interactions. This approach to emotion estimation is intended as a baseline for comparing more complex techniques.

### 5.4.2 Learning methods

The use of learning methods is motivated by the idea of adapting more efficiently to the variety and difficulty of the emotion recognition problem. In this way, we expect a learning method to yield better performance than the one given by the weighted average approach.

As many learning-based techniques, the first step is to extract a set of features that describe the problem. In this specific situation, the number of users $n_u$ is not fixed, but it varies for each user. Due to this, a comprehensive set of features has been manually developed, comprising the information enclosed in $w$ and $E$, as well as several more features. We introduce here a new vector, $w_u$ composed of $n_u$ numbers, where $w_u$ is the number of messages processed by neighbouring user $i$. The complete set of features is described in the Table 9, where $d$ is the number of emotions considered.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Dimensionality</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregated distribution of the users</td>
<td>$d$</td>
<td>Count of the predominant emotion showed by each one of the users connected to the current user</td>
</tr>
<tr>
<td>$n_u$</td>
<td>1</td>
<td>Number of users connected to the current user</td>
</tr>
<tr>
<td>Max, min and mean of $E$</td>
<td>$3d$</td>
<td>Maximum, minimum and average of the values of $E$, column-wise. That is, for each of the considered emotions</td>
</tr>
<tr>
<td>Max, min and mean of $w$</td>
<td>3</td>
<td>Maximum, minimum and average of the vector $w$</td>
</tr>
<tr>
<td>Max, min and mean of</td>
<td>3</td>
<td>Maximum, minimum and average of the vector</td>
</tr>
</tbody>
</table>
Using these features, two different tasks to solve can be considered. The first one, emotion classification, attempts to predict the emotion of a user considering the emotion as a unique label (e.g., represented by a vector \([0 \ 1 \ 0 \ 0]\)). For this, a neural network classifier has been used. The network consists on four hidden layers, with 64, 64, 64 and 32 hidden units each. In total, considering the input and output layers, a 6-layer neural network is used.

For the second type of emotion learning, a multivariate regression problem is considered, where the regressor is trained with the objective of representing the emotions as a distribution rather than as a unique label (e.g., represented by a vector \([0.1 \ 0.05 \ 0.5 \ 0.35]\)). We consider that this task is more difficult than the classification task, as the values to infer are more numerous, and the granularity is higher. For this task, an ensemble of randomised regressor trees has been used, known as Extremely Randomised Trees algorithm [Geurts et al., 2006].

### 5.4.3 Experimentation and results

For the emotion prediction task, several experiments have been conducted on the Weibo dataset [Fan et al., 2014]. This dataset was created by crawling posts and reposts in Weibo from April to September 2010, and each of the posts in the dataset has one of the following labels: anger, disgust, joy and sadness. In order to compute the necessary parameters for the described emotion model, this dataset has been processed, obtaining the aggregation of the weibos sent by each user, and the matrix of relations between users.

Firstly, for the weighted average approach, the experiment has consisted of extracting the required vectors and matrices for each user on the social network and computing the corresponding emotion vector. As this is an unsupervised method, evaluation is done on the whole data set.
Table 10: Emotion classification performances

<table>
<thead>
<tr>
<th>Classification</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
<th>F1Score (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighted average</td>
<td>63.74</td>
<td>61.44</td>
<td>61.35</td>
</tr>
<tr>
<td>Neural Network</td>
<td>61.86</td>
<td>65.97</td>
<td>62.03</td>
</tr>
</tbody>
</table>

In the learning case, similarly, the features have been extracted first. Once a comprehensive set of features is obtained, the data has been divided on train, development and test sets. The training data has been used for learning the hidden parameters of the models (e.g., the internal weights and bias of the neural network), while the development set has been used for learning the hyperparameters of the algorithms (e.g., the number of layers and hidden units on the neural classifier). The test set has been used on the final evaluation of the models. Also, as in the previous experiment, the classification score is the F1-score, and the multivariate regression is the R2-score. These scores are presented in Table 10 for the classification case, and in Table 11 for the regression task.

In the regression task, the results did not achieve the expected performance. This confirms the idea that the multivariate regression task in the emotion domain is difficult, more that in the classification case. For the classification task, both the weighted average and the neural method yield promising results. We expect better results with the combination of these social-oriented features with Natural Language Processing and emotional information.

In comparison to other works, we found similar performance scores for emotion classification in the Weibo dataset. The work presented in [Li et al., 2014] reports F1-scores ranging from 50.40 to 74.93 using text-based approaches; while [Zhao et al., 2012] presents an emoticon-based method that yields a precision of 64.3%. Performance of the proposed approach is comparable to other NLP-based methods whilst using quite distinct data, and so future work combining the methods seems a promising option for improving emotion detection.

### 6 Conclusions and future work

Traditional emotion recognition techniques usually focus on isolation of the individual instead of using its context to enable richer interpretation of its mood. This document exposes the results of task T4.5: Social Context Analysis for Emotion Recognition, which explores the use of Social Context data to improve the emotion recognition for the users of social media platforms. This document presents the Social Context Model used in MixedEmotions project and an overview of Social Context Analysis module to provide the analyses and their results as a web service through
a RESTful API. The complete description of that module will be presented in deliverable D5.4 as part of the project infrastructure.

Moreover, different static and temporal analyses have been performed and explained in this document. The highlights of those experiments have also been included as results.

During the preparation of the experiments in this deliverable, we produced results that can be later used to improve on the experiments. For instance, we generated a social context dataset with the annotated tweets in the Sentiment140\(^4\) dataset, and in dataset of Task 10 (subtask A) of Semeval 2015.

7 References


\(^4\)http://www.sentiment140.com/


Lam, A. J. 2016. Improving Twitter Community Detection through Contextual Sentiment Analysis of Tweets. ACL 2016, 30.


MixedEmotions, 2016. D5.4, Social Semantic Knowledge Graph, Infrastructure and API, final version.


- WSDM '11 Proceedings of the fourth ACM international conference on Web search and data mining.
- Yang, J., & Leskovec, J., 2011. Patterns of temporal variation in online media (pp. 177–186). ACM.
• West, R., Paskov, H. S., Leskovec, J., & Potts, C, 2014. Exploiting Social Network Structure for Person-to-Person Sentiment Analysis CoRR, abs/1409.2