EMOD: An End-to-End Approach for Investigating Emotion Dynamics in Software Development

Krishna Prasad Neupane, Kabo Cheung, and Yi Wang
Rochester Institute of Technology
New York, USA
{kpn3569, kbc5043, yi.wang}@rit.edu

***Abstract*—Emotions are an integral part of human nature. Emotion awareness is critical to any form of interpersonal communication and collaboration, including these in the software development process. Recently, the SE community starts having growing interests in emotion awareness in software development. While researchers have accomplished many valuable results, most extant research ignores the dynamic nature of emotion. To investigate the emotion dynamics, SE community needs an effective approach to capture and model emotion dynamics rather than focuses on extracting isolated emotion states. In this paper, we proposed such an approach–EMOD. EMOD is able to automatically collect project teams' communication records, identify the emotions and their intensities in them, model the emotion dynamics into time series, and provide efficient data management. We developed a prototype tool that instantiates the EMOD approach by assembling state-of-the-art NLP, SE, and time series techniques. We demonstrate the utility of the tool using the `IPython`'s project data on GitHub and a visualization solution built on EmoD. Thus, we demonstrate that EMOD can provide end-to-end support for various emotion awareness research and practices through automated data collection, modeling, storage, analysis, and presentation.**

***Index Terms*—Emotion awareness, emotion dynamics, emotion intensity, software project team, time series database.**

I. INTRODUCTION

Emotions are the subjective matters which express states and opinions of mind and can affect productivity, task quality, and job satisfaction [1], [2]. Its importance in various software engineering activities has increasingly recognized in software engineering literature [3]–[5]. Researchers have proposed multiple methods to identify emotions expressed in the software development process and attempted to identify the relationships between emotion states and various software process and product metrics, e.g., [6]–[9].

One of the critical characteristics of emotion is its dynamic nature [10]. Yet, emotion research in software engineering has for long largely neglected the time dynamic aspects of emotions. As far as our current knowledge, there is no software engineering literature to emotion dynamics into consideration. However, psychology literature has shown that investigating the dynamics of emotions is crucial in emotion awareness research, which is more important than capturing discrete, isolated emotion states [11]–[13]. In software development, for example, a negative emotion state may be not strong enough to indicate unfavorable project progress, but a continuous negative emotion over a few weeks is likely to imply some project problem.

To investigate the dynamics of emotions, the SE research community needs a systematic approach to identify, quantify, model, store, and present dynamic emotion information. In this paper, we present EMOD–an approach that is able to achieve the above goals. Built on state-of-the-art NLP, SE, and time series database techniques, EMOD can continuously collect textual project communication records according to the user-defined data collection time interval, automatically identify the emotions, and calculate the intensities expressed in them. Then, EMOD models the extracted emotions and their intensities into time series, and put them into a time series database for further uses. We instantiated a tool for the EMOD approach and demonstrated how it could be used in visualizing emotion dynamics for emotion awareness in the project team[1].

Compared with the state-of-the-art automated emotion awareness tools in SE, EMOD has several advantages. First, EMOD provides an end-to-end, almost real-time solution for investigating emotion dynamics in software development. Second, EMOD is highly portable and extensible. For each task in it (see Fig. 1), integrating new techniques and replacing the older ones are very convenient. Last but not least, EMOD is not only an approach. Its implementation also provides an infrastructure prototype that can be used in many software engineering research and practice scenarios.

The rest of the paper proceeds as follows. Section II briefly introduces some basics of the psychology of emotion and existing techniques for extracting emotions from the text. The overview of the EMOD approach is presented in section III. Section IV introduces instantiation of EMOD. Discussion, and conclusion are discussed in section V, and VI respectively.

II. BACKGROUND

A. Psychology of Emotions

Emotions are pervasive among humans and have unique values in dealing with fundamental life-tasks [14]. Understanding one's emotions and emotional style build on a vast amount of research in Emotional Psychology and Affective Science. Psychologists have proposed many theories that classify human emotions [15]. According to the discrete emotion theory,

[1] The corresponding source code and data are available at: https://github.com/EmotionDynamic/EmoD

2576-3148/19/$31.00 ©2019 IEEE
DOI 10.1109/ICSME.2019.00038

which is the mainstream view among psychologists, some emotions are considered basic [16], [17]. These basic emotions are assumed to be biologically determined emotional responses whose expression and recognition is fundamentally the same for all individuals regardless of ethnic or cultural differences.

A number of empirical and analytical theories have been proposed on which basic emotions. Ekman [16] in his study of analyzing individual's facial expressions proposed six universal basic emotions: *anger*, *disgust*, *fear*, *joy*, *sadness*, and *surprise*. Although Ekman's basic emotion set is frequently used for emotion mining and modeling, there also exist a wide variety of basic emotions models. For example, in a recent study, Cowen & Keltner [18] from UC Berkeley reported that they identified 27 basic emotions using crowdsourced video analysis of 2,185 videos. In this paper, we use the four of the six basic emotions in Ekman's model. The four emotions are: *joy*, *anger*, *sadness*, and *fear*. We choose it mostly for practical considerations. Most of the state-of-art emotion recognition techniques in NLP focus on these four emotions [19]–[21].

In addition to developing basic emotion models, psychologists also investigated how people's emotions fluctuate across time and argued that studying emotion dynamics may yield more interesting insights related to individual and collective well-beings [13]. In this paper, we will put our focus on emotion dynamics also.

B. Identifying Emotions from Text

Software engineering researchers have developed tools to support automatic emotion extraction. Such tools extract emotion mainly through two ways: (1) explicitly mapping words in the text to some emotion lexicon, or (2) implicitly identifying emotions using machine/deep learning techniques. From several dozens of such learning based techniques, we select the following three, which represent a variety of similar techniques and have the potential to be used in SE scenarios.

- EmoTxt–This is a toolkit for emotion recognition from the text. As far as our current knowledge, it is the first domain-specific tool for software development tasks. EmoTxt contains six binary classifiers. Each classifier uses uni- and bi-grams, emotion lexicon, politeness, sentiment score, and uncertainty to classify textual data related to software development. EmoTxt can detect six emotions in Ekman's basic emotion model. Using a JIRA dataset created by Ortu et al. [20] and a homemade StackOverflow dataset [22], the tool exhibits good performances.
- Senti4SD–It was developed by the same group of researchers who proposed EmoTxt. Senti4SD is a sentiment polarity classifier for software developers' artifacts [8]. The classifier is trained and tested on a gold standard of over 4K posts mined from Stack Overflow and manually annotated with emotion polarity. The novelty of Senti4SD lies on incorporating semantic features through a 600 dimension word embeddings trained on a large Stack Overflow dataset.
- Watson ToneAnalyzer–This is an off-the-shelf solution for detecting emotions and language tones in written text. It is implemented as a RESTful service of IBM Watson Developer Cloud. The service can return results for the following tone IDs: *joy*, *anger*, *fear*, and *sadness* (emotional tones); *analytical*, *confident*, and *tentative* (language tones). The current version uses neural embedding framework to predict emotions [23]. In addition to predicting emotion states, it also provides a confidence score to indicate how confident the algorithm is in detecting such a tone. However, the confidence score does not necessarily represent the intensity of a predicted emotion though they are often correlated.

III. EmoD in A Nutshell

In this section, we discuss the detailed design of EmoD approach. In a single data collection and analysis cycle, EmoD performs five tasks: Data Collection, Data Pre-processing, Emotion Classification, Emotion Intensity Calculation, and Time-series Data Management. Fig. 1 shows the workflow of the approach. To enable the continuous data collection and analysis, EmoD allows the iterations of this five-step process according to user defined time interval. For example, a project team may determine to perform the data collection and analysis on a daily basis or an hourly basis.

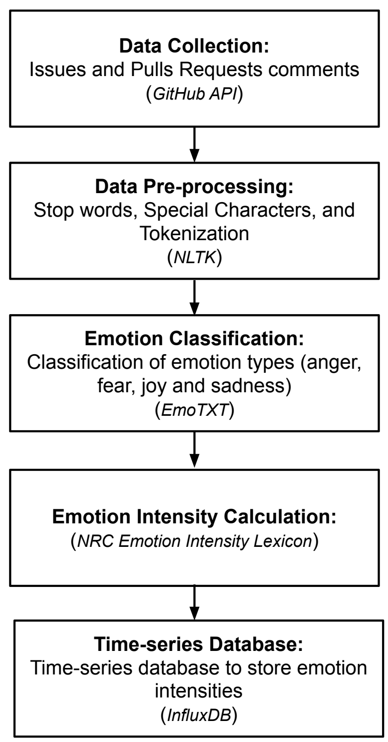

Fig. 1: The workflow of EmoD in a single data collection and analysis cycle.

Now let us have a look at each task in one cycle. The data collection task (*task 1*) is responsible for collecting the text communication record in the predefined time interval. After collecting all new communication records, EmoD performs

TABLE I: A Time-series Database for Emotion Dynamics

Time	Comments	Developer	Developer Role	Anger	Fear	Joy	Sadness
01/09/2018	sorry you're having trouble with this - i think part of the reason is that it's clear what you want to do in trivial examples...	ivanov	Code Contributor	0.23	0.17	0.00	0.00
03/31/2019	next attempt. i kind of stuck to my original recursion but tried to include your suggested checks to avoid revisiting of already browsed objects...	daharn	Other Contributor	0.00	0.30	0.00	0.00
01/13/2018	i would start by looking at related project has you are not the only one who are interested in it ; for example : - https://multithreaded.stitchfix.com/blog/2017/07/26/nodebook/ i know there are more around. it is a complex enough problem that a team of interested people are more change to do something that work....	Carreau	Code Contributor	0.00	0.00	0.00	0.11

the data pre-processing task (*task 2*). Then, the cleaned text is classified using proper sentiment/emotion classification tools (*task 3*). The next step is calculating the recognized emotions' intensities (*task 4*). Once the calculation finished, the results are appended to the time series database and for further use (*task 5*).

Please note that, *task 3* and *task 4* may be combined together if using some techniques that can directly computing emotion intensities. We separated them because there is no such technique in SE research. Besides, in Fig. 1, we also show one potential technique as an example for each task. However, the techniques used when implementing EMOD are not restricted to them. There are many alternative techniques. These techniques listed in Fig. 1 are the techniques we used in the prototype tool, which will be introduced in the next section.

IV. INSTANTIATION OF EMOD

We instantiate a prototype tool that encapsulates the EMOD approach. The tool contains five major components. Each of them corresponds to one of EMOD's major task. The tool uses the data from GitHub projects. We now will discuss the detailed implementation of it. We will use the `IPython` project hosted on GitHub as an example. All source code and data are made available at: https://github.com/EmotionDynamic/EmoD.

- *Data Collection.*
  The tool uses comments of issues and pull requests of GitHub projects as the main source of a project's communication records. The data collection was implemented using GitHub API[2]. For the project `IPython`, we collected all data from January 1st, 2018, and continue sensing the new communication records on a daily basis. By the date of June 1st, 2019, we have data generated by 10 code contributors and 340 other contributors. The data collection is still ongoing every day. At the midnight (12:00 AM, UTC-4) of each day, the tool will automatically start, download all new issue and pull request comments generated in the last 24 hours, and feed them into the following tasks.
- *Data Pre-Processing.*
  Upon the finish of the daily data collection, the newly collected comments are preprocessed following the standard protocol. This component is implemented based on the `NLTK`[3], the pre-processing library for `Python`. This includes the removal of stop words, special characters, and tokenization. After cleaning and tokenizing text, the preprocessed texts are ready for the emotion classification.
- *Emotion Classification with EmoTxt.*
  In this step, we used the open source emotion detection tool–EMOTXT [24] to classify the comments to different emotions. It has been introduced in Section II-B. Since the granularity of the emotion analysis is on a daily basis, we grouped an individual's comments in one day together. For example, if author *daharn* has two comments in a specific day then it goes in a single comments column of the time-series database and its corresponding emotional intensities are computed and represented for a whole day. Then, EMOTXT determines if the text expresses any of the four emotions (anger, fear, joy, and sadness). After labeling the text with proper emotions, the intensity of each emotion is calculated in the next step.
- *Emotion Intensity Calculation based on NRC Emotion Intensity Lexicon.*
  As we mentioned in Section III, there is no mature tool to identify the emotion intensities in software engineering research directly. We had to develop a method by ourselves. In the last step, we already identified the emotions in a text; we can simply use lexicon-based methods to identify the

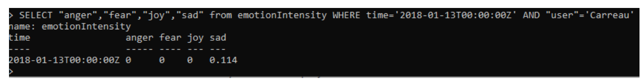

Fig. 2: An example of querying the time series database for `IPython`.

[2] https://developer.github.com/v3/

[3] https://www.nltk.org/

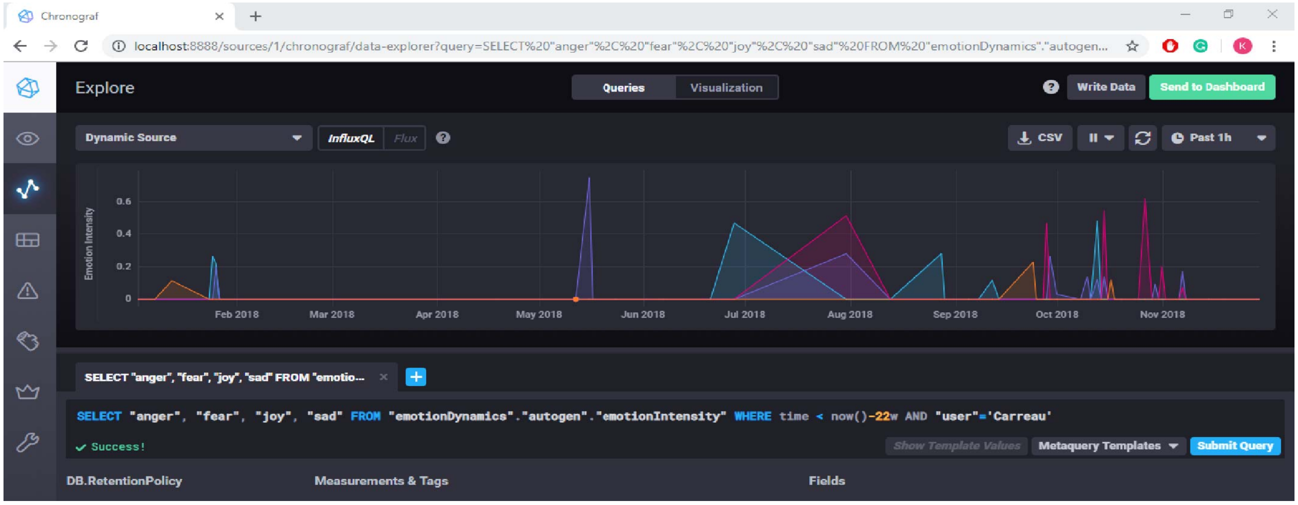

Fig. 3: A screenshot of the emotion dynamics visualization tool built on the chronograf. The developer *Carreau*'s emotion dynamics in the year 2018 are displayed.

specific "words" related to these emotions and compute the emotion intensities with these words' emotion intensities. To do so, we use the NRC emotion intensity lexicon. It contains 5,814 unigram words with corresponding emotion intensity scores [25]. In which, 1483 words are anger, 1765 words are fear, 1298 words are sadness, and 1268 words are a joy. The range of emotional intensities is between 0 and 1. Intensities close to 0 represent weak emotion, and those close to 1 represent strong emotion. For each text, we followed the momentum emotion theory [26], thus simply using the strongest word's intensity in the text to represent the text's emotion intensity.

- *Data Modeling and Storage.*
  Obviously, the emotion dynamics consist of the computed emotion intensities for each individual in a project and grow every day. Thus, we can view them as a time series. Several databases are specially designed and optimized for time series data. We used the InfluxDB[4] which stores time-stamped dataset and provides efficient reading-writing capabilities. In our design, each project will have its database in a collection of multiple projects' database. In each database, a record has time-stamp, developer name, developer role (*code contributor* and *other contributor*), comments, emotional intensities: anger, fear, joy, and sadness as an attribute of a database table. Three example records in the database's table are shown in Tab. I.
  With the functionality offered by the InfluxDB, a project's database can efficiently support various operations. For example, a user, who could be either a researcher or a project member, can retrieve emotion data using its SQL-like query language. It is very convenient to find emotional intensities, developers types based on their roles, and emotion types for the active developers within a specific time frame. Fig. 2 provides such an example. In this example, the query `SELECT` is used with `WHERE` condition to select emotional intensities of only *Carreau* developer for the specific time. In addition, to support basic queries, InfluxData, which InfluxDB is a part of, is a time series database ecosystem that consists of many tools which are potentially used to build sophisticated applications without too much extra effort. In the next section, we will show dynamic emotion visualizations built with tools in the InfluxDB ecosystem.

V. Discussion

In this section, we breifly discuss some potential applications of the EMOD approach and its future evolution.

A. *Applications of* EMOD

EMOD and its implementation have many potential uses in supporting research and practice related to emotion awareness. For example, researchers can extract the features of the emotion dynamics and link them with project outcomes; practitioners can build emotion awareness tools that monitor team's emotion changes and perform early interventions, and so on. As an example , we use a simple visualization tool to demonstrate the feasibility of these potential use cases.

For each project, EMOD tool produces an time series database, which can be directly used to visualize the emotion dynamics of each team members based on records of corresponding emotion intensities in the database. To visualize the emotion dynamics, we employed chronograf[5] tool. Similar to InfluxDB, chronograf is also a part of the InfluxData ecosystem; thus, it can directly integrate with and operate any influxDB database. Hence, the visualizations can be directly generated from influxDB queries. Fig. 3 is the screenshot of the visualization tools' interface built on chronograf. It displays the emotion dynamics of the most active developer

[4] https://www.influxdata.com/products/influxdb-overview/

[5] https://www.influxdata.com/time-series-platform/chronograf/

Carreau of the `IPython` project in pulls and issues comments. This figure includes four emotion types (*anger:* with the green color line, *fear:* with the purple color line, *joy:* with the red color line and *sadness:* with the orange color line) and their intensities for each day of the whole year of 2018. During the days of month May, the most active developer *Carreau* seems quite fear than other days. Similarly, there is a lot of variation in emotions types and emotion intensities during the days of July and August.

B. Future Evolution of EMOD

The EMOD approach and the corresponding tool, though in their early stage of iteration, have shown some promises. In the future, we plan to further improve the approach and the tool, for example, we may combine the *task 3* and *task 4* together by adapting NLP techniques that can provide reliable emotion intensities. We also plan to make the tool support data sources other than GitHub. Currently, we only applied it into one project–IPython. We will try to apply it to a set of open source projects, and contribute the databases of these project as the data infrastructure for emotion awareness research in the software engineering community.

VI. CONCLUSION

In this paper, we proposed an approach named EMOD for supporting emotion awareness research and practice in software development. EMOD is able to automatically collect project teams' communication records, identify the emotions and their intensities in them, model the emotion dynamics into time series, and provide efficient data management. We developed a prototype tool that instantiates the EMOD approach by assembling state-of-the-art NLP, SE, and time series techniques. We demonstrate the utility of the tool using the `Ipython`'s project data on GitHub and a visualization solution built on EmoD. Thus, we demonstrate that EMOD can provide end-to-end support for various emotion awareness research and practices through automated data collection, modeling, storage, analysis, and presentation.

ACKNOWLEDGEMENT

This work is partially supported by National Science Foundation under awards CCF-1757680 and IIS-1850067.

REFERENCES

[1] R. Feldt, L. Angelis, R. Torkar, and M. Samuelsson, "Links between the personalities, views and attitudes of software engineers," *Information and Software Technology*, vol. 52, no. 6, pp. 611–624, 2010.

[2] M. R. Islam and M. F. Zibran, "Towards understanding and exploiting developers' emotional variations in software engineering," in *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, pp. 185–192, IEEE, 2016.

[3] P. Dewan, "Towards emotion-based collaborative software engineering," in *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, pp. 109–112, IEEE, 2015.

[4] A. Fountaine and B. Sharif, "Emotional awareness in software development: Theory and measurement," in *2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion)*, pp. 28–31, May 2017.

[5] A. Kołakowska, A. Landowska, M. Szwoch, W. Szwoch, and M. R. Wróbel, "Emotion recognition and its application in software engineering," in *2013 6th International Conference on Human System Interactions (HSI)*, pp. 532–539, IEEE, 2013.

[6] E. Guzman, D. Azócar, and Y. Li, "Sentiment analysis of commit comments in github: an empirical study," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, pp. 352–355, ACM, 2014.

[7] M. R. Islam and M. F. Zibran, "Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text," *Journal of Systems and Software*, vol. 145, pp. 125–146, 2018.

[8] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment polarity detection for software development," *Empirical Software Engineering*, vol. 23, no. 3, pp. 1352–1382, 2018.

[9] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, "Sentiment analysis for software engineering: How far can we go?," in *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*, pp. 94–104, IEEE, 2018.

[10] P. Kuppens and P. Verduyn, "Emotion dynamics," *Current Opinion in Psychology*, vol. 17, pp. 22 – 26, 2017. Emotion.

[11] A. K. Anderson, "Affective influences on the attentional dynamics supporting awareness.," *Journal of experimental psychology: General*, vol. 134, no. 2, p. 258, 2005.

[12] J. T. MacCurdy, *The psychology of emotion: morbid and normal*. Routledge, 2013.

[13] M. Houben, W. Van Den Noortgate, and P. Kuppens, "The relation between short-term emotion dynamics and psychological well-being: A meta-analysis.," *Psychological bulletin*, vol. 141, no. 4, pp. 901–930, 2015.

[14] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[15] A. Ortony and T. J. Turner, "What's basic about basic emotions?," *Psychological review*, vol. 97, no. 3, p. 315, 1990.

[16] P. Ekman, "Are there basic emotions?," 1992.

[17] P. Ekman, *Emotions revealed: Recognizing faces and feelings to improve communication and emotional life*. Macmillan, 2007.

[18] A. S. Cowen and D. Keltner, "Self-report captures 27 distinct categories of emotion bridged by continuous gradients," *Proceedings of the National Academy of Sciences*, vol. 114, no. 38, pp. E7900–E7909, 2017.

[19] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "Semeval-2018 task 1: Affect in tweets," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 1–17, 2018.

[20] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli, "Are bullies more productive?: empirical study of affectiveness vs. issue fixing time," in *Proceedings of the 12th Working Conference on Mining Software Repositories*, pp. 303–313, IEEE Press, 2015.

[21] C. Baziotis, N. Athanasiou, A. Chronopoulou, A. Kolovou, G. Paraskevopoulos, N. Ellinas, S. Narayanan, and A. Potamianos, "Ntua-slp at semeval-2018 task 1: predicting affective content in tweets with deep attentive rnns and transfer learning," *arXiv preprint arXiv:1804.06658*, 2018.

[22] N. Novielli, F. Calefato, and F. Lanubile, "A gold standard for emotion annotation in stack overflow," in *Proc. of the 15th International Conference on Mining Software Repositories (MSR)*, pp. 14–17, 2018.

[23] Z. Liu, A. Xu, Y. Guo, J. U. Mahmud, H. Liu, and R. Akkiraju, "Seemo: A computational approach to see emotions," in *Proc. of the 2018 CHI Conference on Human Factors in Computing Systems (CHI)*, pp. 364:1–364:12, 2018.

[24] F. Calefato, F. Lanubile, and N. Novielli, "Emotxt: a toolkit for emotion recognition from text," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pp. 79–80, IEEE, 2017.

[25] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word-emotion association lexicon," vol. 29, no. 3, pp. 436–465, 2013.

[26] E. Eldar, R. B. Rutledge, R. J. Dolan, and Y. Niv, "Mood as representation of momentum," *Trends in cognitive sciences*, vol. 20, no. 1, pp. 15–24, 2016.