Details on data processing

We followed most of the same processes described by *Ahne et al.* in their Supplementary online material 2.[1]

Data collection

We accessed the API Standard v1.1 by applying for a Twitter Developer account.[2] We connected to Twitter's API using the Python library Tweepy and streamed the keywords to collect tweets together with users' metadata (attributes provided on user's profile, such as user screen name, user location, user description.).[3]

Data representation

We used FastText implementation in the Gensim package to process each word into a vector to extract meaningful semantic relationships.[4] The average of each tweet's word vector representations was then used to model it and similarities in their semantics were analyzed.

Geolocation process

We aimed to keep only geolocated tweets for this study. To do so, we first looked at users' locations, provided by users themselves in their public profile, and removed all tweets without such location. We then grouped all locations and manually detected which ones were fake in order to exclude it (e.g. "the Internet", "in Hell"). We also replaced all contractions to full words (eg. "CA, USA" to "California, United States of America") to facilitate geolocation by the Python package geograpy4.[5] We then applied the package to the different grouped locations in order to access the latitude

and the longitude of the location associated with each tweet. If the determined place was a country, latitude and longitude were identified in the center of the country. We evaluated the overall precision of this geolocation step as 85%.

Tweets translation

In order to apply unique classifiers to all tweets, we translated all tweets originally not written in English to English using the Python package deep-translator, a free and unlimited python API for Google Translate.[6]

Personal content classifier / Jokes classifier

We developed two classifiers that aimed to filter out institutional tweets and jokes in order to keep only tweets with personal content and not jokes or irony about diabetes. A tweet was considered as personal if the user expressed his feelings and own experiences, dealing with his own diabetes or a relative one. A tweet was considered as a joke/irony if diabetes was used as an insult or with a sugar-related joke. To train these two classifiers, three authors (AA, CB, GF) manually labelled 1648 randomly chosen tweets for the personal classifier and 1398 for the jokes one. We used *Bidirectional Encoder Representations from Transformers* (BERT), a machine learning technique for natural language processing pre-training developed by Google.[7] More precisely, we applied *BERTweet*, a pre-trained model for English Tweets.[8] The overall accuracy of the personal content classifier was 88% and the accuracy from the jokes classifier was 94%.

Gender and Type of diabetes classifier

Following the scripts by *Ahne et al.*, we trained classifiers to predict gender (male, female, unknown) and type of diabetes (type 1, type 2, unknown) from each user.[1] Three authors (AA, CB, GF) manually labelled 1670 tweets from different regions of the world to better match our dataset. A SVM was trained and reached an accuracy of 86% for the gender classifier and 74% for the type of diabetes classifier.

Topic extraction

All tweets are represented via their word vector representations. Then, a K-means algorithm was applied to the tweets in each region. To define the optimal number of clusters *k*, the silhouette score average for *k* between 4 and 24 was calculated and silhouette analysis applied.[9]

Emotion classifier

Initially, two authors (CB, GF) labelled 1000 randomly chosen tweets according to the main emotion in the tweet text. In order to increase the accuracy of our classifier, we combined our dataset with online labelled datasets of emotions which led to the extension of 47,000 tweets from Github and Kaggle.[10–12] We then trained a Calibrated Linear Support Vector classifier to predict the probability of a tweet belonging to each of the four emotions.[13] We applied this classifier to all tweets to predict the probability of a tweet belonging to each of the four emotions.

This led to the creation of a dataset of 47,926 texts labelled as one of the following emotions: joy, anger, fear or sadness. We based our classifier on the following scripts: https://thecleverprogrammer.com/2021/02/19/text-emotions-detection-with-machine-learning/. We calibrated the classifier in order to predict the probability of a

tweet to belong to each of the 4 classes.[14] The calibrated SVM was trained and reached an accuracy of 80.56% (precision to predict joy only: 88%, precision to predict fear only: 94%, precision to predict anger only: 93%, precision to predict sadness only: 89%).

Details

Python (V.3.6) with the packages scikit-learn (machine learning algorithms and data preprocessing methods), gensim (text processing, word representation), and statsmodels (module for the estimation of many different statistical models) were exploited for the analysis.[15–17] Tableau (2020.4) and OpenStreetMap 2021 were used to visualize the data.

Additional references

1. Ahne A, Orchard F, Tannier X, Perchoux C, Balkau B, Pagoto S, et al. Insulin pricing and other major diabetes-related concerns in the USA: a study of 46 407 tweets between 2017 and 2019. BMJ Open Diabetes Res Care [Internet]. 2020 Jun;8(1). Available from: http://dx.doi.org/10.1136/bmjdrc-2020-001190
2. Website [Internet]. Available from: “Data Dictionary: Standard v1.1.” n.d. Twitter Developer Platform. https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/user.
3. Roesslein J. Tweepy: Twitter for python. URL: https://github com/tweepy/tweepy. 2020;484.
4. fastText [Internet]. [cited 2021 Oct 12]. Available from: https://fasttext.cc/index.html
5. ThomasShih. GitHub - ThomasShih/geograpy4: Extract countries, regions and cities from a URL [Internet]. [cited 2021 Oct 13]. Available from: https://github.com/ThomasShih/geograpy4
6. Welcome to deep_translator’s documentation! — deep_translator documentation [Internet]. [cited 2021 Oct 12]. Available from: https://deep-translator.readthedocs.io/en/latest/

7. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Internet]. 2018 [cited 2021 Oct 12]. Available from: http://arxiv.org/abs/1810.04805

8. Nguyen DQ, Vu T, Nguyen AT. BERTweet: A pre-trained language model for English Tweets [Internet]. 2020 [cited 2021 Oct 12]. Available from: http://arxiv.org/abs/2005.10200

9. Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis [Internet]. Vol. 20, Journal of Computational and Applied Mathematics. 1987. p. 53–65. Available from: http://dx.doi.org/10.1016/0377-0427(87)90125-7

10. Praveen. Emotions dataset for NLP [Internet]. [cited 2021 Oct 13]. Available from: https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp

11. Jcharis. GitHub - Jcharis/end2end-nlp-project: End 2 End NLP Project with Python [Internet]. [cited 2021 Oct 13]. Available from: https://github.com/Jcharis/end2end-nlp-project

12. Kharwal A. Text Emotions Detection with Machine Learning [Internet]. 2021 [cited 2021 Oct 13]. Available from: https://thecleverprogrammer.com/2021/02/19/text-emotions-detection-with-machine-learning/

13. Tang Y. Deep Learning using Linear Support Vector Machines [Internet]. arXiv [cs.LG]. 2013. Available from: http://arxiv.org/abs/1306.0239

14. sklearn.calibration.CalibratedClassifierCV [Internet]. [cited 2021 Oct 13]. Available from: https://scikit-learn/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html

15. Rehurek R, Sojka P. Gensim--python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic. 2011;3(2).

16. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res. 2011;12:2825–30.

17. Seabold S, Perktold J. Statsmodels: Econometric and Statistical Modeling with Python [Internet]. Proceedings of the 9th Python in Science Conference. 2010. Available from: http://dx.doi.org/10.25080/majora-92bf1922-011