The language of mental health problems in social media

George Gkotsis1,†, Anika Oellrich1,†, Tim Hubbard2, Richard JB Dobson1,3
Maria Liakata4, Sumithra Velupillai1,5, Rina Dutta1
1King’s College London, IoPPN, London, SE5 8AF, UK; e-mail: george.gkotsis@kcl.ac.uk
2King’s College London, Guy’s Hospital, London, SE1 9RT, UK
3Farr Institute of Health Informatics Research, London, WC1E 6BT, UK
4University of Warwick, Department of Computer Science, Warwick, CV4 7AL, UK
5School of Computer Science and Communication, KTH, Stockholm
†Authors contributed equally to this work.

Supplement

Determining relevant subreddits

In order to identify relevant subreddits, a heuristic method for searching on free text was applied. More specifically, we searched across the complete downloaded reddit archive for text in posts or comments that match against certain keywords (see Table 1 for the complete list of 134 keywords). This keyword list was expert-curated and is released as part of the PHEME project1 (Kolliakou et al., 2015). The result set was grouped by subreddit and sorted by size. We then visited the top subreddits and curated a preliminary list of relevant subreddits. Based on this list, we fetched the complete data for these subreddits. The final list was compiled after eliminating subreddits that contained fewer than 5,000 communications. Table 2 displays this final list, together with an explanation of the purpose of this subreddit as provided on Reddit.

<table>
<thead>
<tr>
<th>Table 1: List of keywords used for mining relevant subreddits.</th>
</tr>
</thead>
</table>

Generation of language features for posts and comments from Reddit

To generate the language features, we used a Python implementation. All relevant scripts are available on our github repository2. The core Python modules used for our analysis include Pandas (data analysis)3, spaCy (NLP)4, seaborn (statistical data visualisation)5 and scikit-learn (machine learning)6. We briefly present some details behind the implementation.

Regarding the language features, we used spaCy for analysing the free text. The library provides a parse tree which contains syntactic information about the token-words in each sentence. These tokens were used to identify the pronouns (i.e. pronouns, definite articles), verbs and verb phrases (sequences of verbs), as well as nouns and noun chunks (sequences of nouns). Furthermore, the tree for each sentence was used in order to calculate its height. Similarity between adjacent sentences (see LF10 and LF11 in Table 2, manuscript) was implemented using a cosine similarity function.

Determination of discriminability of subreddits based on subreddit-specific vocabularies

Our approach to determine language homogeneity across subreddits, examines the number of occurrences of unigrams (words) and bigrams (sequences

1http://www.pheme.eu/
2https://github.com/gkotsis/reddit-mental-health
3http://pandas.pydata.org/
4https://spacy.io/
5https://stanford.edu/~mwaskom/software/seaborn/
6http://scikit-learn.org/stable/
of 2 words) in each document. The numbers of occurrences for uni- and bigrams were the input for a Naive Bayes classifier. Because the sizes (in terms of number of communications) of the subreddits varied significantly (imbalanced datasets), we aimed to set up the classification datasets in a uniform and consistent way. The balanced dataset is produced by randomly selecting 5k posts from each subreddit, or the maximum number of posts from the smallest of the two (whichever is the highest). The overall accuracy of this classification task (number of correctly classified instances over total instances) is the average accuracy produced from 10 folds. Hence, we consider the produced mean accuracy to represent a language proximity feature between two subreddits: a high accuracy score means that the two subreddits are distinctive in terms of language (low proximity), whereas a low accuracy score depicts a shared vocabulary across two subreddits (high proximity).

Assessment of affection in subreddits

To determine sentiment in post of the relevant subreddit, we used two specific methods: LabMT (Dodds et al., 2011) and AFINN (Nielsen, 2011). For both methods, we used the list of words as published with the respective paper. The result of this normalisation was that words with low score (e.g. “suicide”, “death”) received positive score. The score of each document was calculated by taking the sum of the values of each word found. The final score value was then further normalised by dividing the sum by the square root of the number of words found. For the AFINN feature, the word list already contained positive and negative values for the so-called “valence” of each word and was implemented in a similar approach.

References


A Kolliakou, G Gkotsis, M Ball, D Chandran, R Zubiaga, R Dutta, and R Stewart. 2015. D7.2.2.2 - annotated corpus - final version public deliverable, pheme project (fp7-ict-611233).