# Exploring the Performance of Baseline Text Mining Frameworks for Early Prediction of Self Harm Over Social Media

Tanmay Basu[1,2,3], Georgios V. Gkoutos[1,2,3,4,5]

[1]*Center for Computational Biology, University of Birmingham, UK*
[2]*Institute of Translational Medicine, University Hospitals Birmingham, UK*
[3]*MRC Health Data Research UK (HDR UK), Midlands Site, Birmingham, UK*
[4]*NIHR Experimental Cancer Medicine Centre, Birmingham, UK*
[4]*NIHR Surgical Reconstruction and Microbiology Research Centre, Birmingham, UK*

## Abstract

The Task 2 of CLEF eRisk 2021 challenge focuses on early prediction of self-harm based on sequentially processing pieces of text over social media. The workshop has organized three tasks this year and released different corpora for the individual tasks and these are developed using the posts and comments over Reddit, a popular social media. The text mining group at Center for Computational Biology in University of Birmingham, UK has participated in Task 2 of this challenge and submitted five runs for five different text mining frameworks. The paper explore the performance of different text mining techniques for early risk prediction of self-harm. The techniques involve various classifiers and feature engineering schemes. The simple bag of words model and the Doc2Vec based document embeddings have been used to build features from free text. Subsequently, ada boost, random forest, logistic regression and support vector machine (SVM) classifiers are used to identify self-harm from the given texts. The experimental analysis on the test corpus show that the SVM classifier using the conventional bag of words model outperforms the other methods for identifying self-harm. This framework achieves best score in terms of precision among all the submissions of eRisk 2021 challenge for identifying self harm over social media.

## Keywords

identification of self-harm, text classification, information extraction, text mining

## 1. Introduction

Early risk prediction is a new research area potentially applicable to a wide variety of situations such as identifying people with risk of suicidal attempts over social media. Online social platforms allow people to share and express their thoughts and feelings freely and publicly with other people [1]. The information available over social media is a rich source for sentiment analysis or inferring mental health issues [2]. The CLEF eRisk 2021 shared task focuses on

*CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania*

welcometanmay@gmail.com (T. Basu); g.gkoutos@bham.ac.uk (G. V. Gkoutos)
https://www.birmingham.ac.uk/staff/profiles/cancer-genomic/basu-tanmay.aspx (T. Basu);
https://www.birmingham.ac.uk/staff/profiles/cancer-genomic/gkoutos-georgios.aspx (G. V. Gkoutos)
0000-0001-9536-8075 (T. Basu); 0000-0002-2061-091X (G. V. Gkoutos)

early prediction of self harm over social media. The main goal of eRisk 2021 challenge is to instigate discussion on the creation of reusable benchmarks for evaluating early risk detection algorithms by exploring issues of evaluation methodology, effectiveness metrics and other processes related to the creation of test collections for early detection of self harm [3]. The workshop has organized three tasks this year and released different corpora for the individual tasks and these are developed using the posts and comments over Reddit, a popular social media [3]. However, we have participated in task 2, which focuses on early prediction of self-harm over social media.

Suicide ranks among the leading causes of death around the world [4, 5]. Self harm is a type of suicide attempts that leads to death most of the times. It is associated with a number of adverse outcomes, and is strongly associated with future suicide [6, 7]. The World Health Organization has recommended that member states develop self-harm monitoring systems as part of their suicide prevention efforts [5]. However, it is generally difficult to identify self harm from the early symptoms. The treatment for these diseases can be started on time, if the alarming symptoms are diagnosed properly. The recent research has focused on identifying self-harm from medical records using machine learning and text mining approaches [8, 4, 5]. The CLEF eRisk Lab have been organizing shared task on early risk prediction of self harm over social media since 2019 [3, 9, 10].

Burdisso et al. developed a text classification framework for early risk detection based on confidence between the concepts and categories which performs very well for early prediction of self harm in eRisk 2019 [11, 12]. The BiTeM group at eRisk 2019 explored different baseline approaches including convolutional neural network, bag words model and SVM classifier for early prediction of self harm [13]. Martınez et al. used different BERT based classifiers which were trained specifically for early prediction of self harm in eRisk 2020 [14]. They used a variety of pretrained models including BERT, DistillBERT, RoBERTa and XLM-RoBERTa and finetuned these models on various training corpora from Reddit, which they created [14, 10]. Ageitos et al. implemented a machine learning approach using textual features and SVM classifier for early prediction of self harm in eRisk 2020 [15]. In order to extract relevant features, they followed a sliding window approach that handles the last messages published by any given user. The features considered a wide range of variables, such as title length, words in the title, punctuation, emoticons, and other feature sets obtained from sentiment analysis, first person pronouns and words of non-suicidal self-injury [15, 10].

In this paper, different text mining frameworks have been developed to identify self harm over social media data released as part of eRisk 2021 shared task. The aim is to train a machine learning classifier using the given training corpus to classify individual documents of the test corpus. The performance of a text classification technique is highly dependent on the potential features of a corpus. Therefore the performance of different classifiers have been tested following different feature engineering techniques. The conventional bag of words model [16] and the Doc2Vec based deep learning model [17, 18] have been used to generate features from the given corpora. We have explored two different term weighting schemes of the bag of words model, viz., term frequency and inverse frequency (TF-IDF) based term weighting scheme [19] and entropy based term weighting scheme [20, 21].

Subsequently, the performance of ada boost [22], logistic regression [23], random forest [24] and support vector machine [25] classifiers have been reported using the bag of words features and the Doc2Vec based features individually on the training corpus following 10 fold cross validation. Therefore, the best five frameworks are chosen based on their performance on the training corpus in terms of fmeasure and subsequently they have been implemented on the test corpus. The experimental results show that the support vector machine classifier using TF-IDF based term weighting scheme outperforms the other frameworks on the test corpus in terms of precision. Moreover, this framework achieves the best precision score among all the runs submitted in Task 2 of the eRisk 2021 shared task.

The paper is organized as follows. The proposed text mining frameworks are explained in section 2. Section 3 describes the experimental evaluation. The conclusion is presented in section 4.

## 2. Proposed Frameworks

Various text mining frameworks have been proposed here to identify the documents that indicate the risk of self harm from the given corpus. The documents of the corpus are released in XML format. Each XML document contains the posts or comments of a Reddit user over a period of time with the corresponding dates. We have extracted the posts or comments from the XML documents and ignored the other entries. Therefore the corpus used for experiments in this article contain only the free texts related to different posts over Reddit for individual users. Different types of features are considered to build the proposed frameworks to identify self harm using state of the art classifiers.

### 2.1. Feature Engineering Techniques

There are different feature engineering techniques exist in the literature of text mining. We have considered both raw text features and semantic features in the proposed methods.

#### 2.1.1. Bag Of Words Features

The text documents are generally represented by the bag of words model [19, 16, 26]. In this model, each document in a corpus is generally represented by a vector, whose length is equal to the number of unique terms, also known as vocabulary [16].

Let us denote the number of documents of the corpus and the number of terms of the vocabulary by $n$ and $m$ respectively. Number of times the $i^{th}$ term $t_i$ occurs in the $j^{th}$ document is denoted by $tf_{ij},\ i = 1, 2, ..., m;\ j = 1, 2, ..., n$. The documents can be efficiently represented following the vector space model in most of the text mining algorithms [19]. In this model each document $d_j$ is considered to be a vector $\vec{d_j}$, where the $i^{th}$ component of the vector is say, $W_{ij}$, i.e., $\vec{d_j} = (W_{1j}, W_{2j}, ..., W_{mj})$. There are different term weighting schemes in the literature to represent the weights of a document vector. However, we have used the following two term weighting schemes to represent the bag of words, which are widely used in the literature [26, 21, 27, 28].

The conventional term weighting scheme is known as term frequency and inverse document frequency or *TF-IDF*. Document frequency, say, $df_i$ is the number of documents in which the $i^{th}$ term appears. Inverse document frequency determines how frequently a term occurs in a corpus and it is defined as $idf_i = log(\frac{n}{df_i})$. The weight of the $i^{th}$ term in the $j^{th}$ document, denoted by $W_{ij}$, is determined by combining the term frequency with the inverse document frequency as follows:

$$W_{ij} = tf_{ij} \times idf_i = tf_{ij} \times log(\frac{N}{df_i}),\ \ \forall\, i = 1, 2, ..., m \text{ and } \forall\, j = 1, 2, ..., n$$

The entropy based term weighting technique is used by many researchers to form term-document matrix from free text data [20, 21]. This method reflects the assumption that the more important term is the more frequent one that occurs in fewer documents, taking the distribution of the term over the corpus into account [21]. Thus the weight of a term $t_i$ in the $j^{th}$ document, denoted by $W_{ij}$, is determined by the entropy based technique[1] [21] as follows:

$$W_{ij} = \log(tf_{ij} + 1) \times \left(1 + \frac{\sum\limits_{j=1}^{n} P_{ij} \log P_{ij}}{\log(n+1)}\right), \quad \text{where,} \quad P_{ij} = \frac{tf_{ij}}{\sum\limits_{j=1}^{n} tf_{ij}}$$

The term-document matrices developed by the bag of words models are generally sparse and high dimensional. The same may have adverse affect on the quality of the classifiers. Hence significant terms related to different categories of a corpus are to be determined. Many term selection techniques are available in the literature. The term selection methods rank the terms in the vocabulary according to different criterion function and then a fixed number of top terms forms the resultant set of features. A widely used term selection technique is $\chi^2$-statistic [26] and this is used in the proposed frameworks. We have considered different number of terms generated by $\chi^2$-statistic and evaluated the performance of individual classifiers using these set of terms on the training corpus. The best set of terms are used for experiments on the test corpus.

#### 2.1.2. Doc2Vec Based Features

A Doc2Vec model can express a document as a vector. We can evaluate semantic similarity between two documents by comparing the corresponding vectors. Doc2Vec model is based on Word2Vec model [29], which expresses a word as a vector. In a vector space produced by a Word2Vec model, two words which are similar in meaning correspond to two vectors which are close to each other [30]. Furthermore, the relationship among words is consistent throughout vector operations, e.g., “king - man + woman = queen” [30]. Doc2Vec is an extension of Word2Vec to learn document level embeddings [17, 18]. Its algorithm is implemented in

[1] https://radimrehurek.com/gensim/models/logentropy_model.html

Gensim[2], a Python library. The Doc2Vec model is trained on the training corpus to generate the embeddings from individual documents. The features of the test documents are inferred from these embeddings learned over the training corpus.

### 2.2. Text Classification Techniques

Different text classification methods have been used to identify self harm in the given test corpus using bag of words features and features generated by Doc2Vec model. The proposed frameworks are developed using Ada Boost (AB), Logistic Regression (LR), Random Forest (RF) and Support Vector Machine (SVM) classifiers.

SVM is widely used for text classification [25]. The linear kernel is recommended for text classification as the linear kernel performs well when there is a lot of features [31]. Hence linear SVM is used in the experiments.

Random Forest is an ensemble of decision tree classifier, which is trained with the bagging method. The general idea of the bagging method is that a combination of learning models increases the overall result. It has shown good results for binary class text classification problems [24]. We have used random forest classifier using gini index as the measure of the quality of a split.

Logistic regression performs well for two class classification problem [23]. We have implemented logistic regression using LibLinear, a library for large-scale linear classification [31].

The Ada boost algorithm is an ensemble technique, which can combine many weak classifiers into one strong classifier [22]. This has been widely used for binary classification problems [32].

## 3. Experimental Evaluation

### 3.1. Experimental Setup

We have submitted five runs following five different frameworks. The overview of the runs are given in Table 1. We have explored the performance of different feature engineering techniques and the classifiers following 10 fold cross validation method on the training corpus and have chosen five best frameworks to be tested on the test corpus. The performance of the proposed frameworks are evaluated by using Precision, Recall, Fmeasure, Early Risk Detection Error (ERDE), $Latency_{TP}$ and speed [3, 33]. These evaluation techniques are described in the overview paper of eRisk 2021 shared task [3, 33]. AB, LR, RF and SVM classifiers are implemented in Scikit-learn[3], a machine learning tool in Python [34]. Doc2Vec is implemented in Gensim[4], a deep learning tool in Python.

[2] https://radimrehurek.com/gensim/models/doc2vec.html#gensim.models.doc2vec.Doc2Vec

[3] http://scikit-learn.org/stable/supervised_learning.html

[4] https://radimrehurek.com/gensim/models/doc2vec.html#gensim.models.doc2vec.Doc2Vec

**Table 1**
Overview of Five Different Runs

| Runs | Frameworks | Number of Features |
|---|---|---|
| Birmingham 0 | Entropy Based Features and SVM Classifier | 3000 |
| Birmingham 1 | Entropy Based Features and RF Classifier | 3000 |
| Birmingham 2 | TF-IDF Based Features and SVM Classifier | 2500 |
| Birmingham 3 | Doc2Vec Based Features and AB Classifier | 50 |
| Birmingham 4 | Doc2Vec Based Features and RF Classifier | 50 |

**Table 2**
Performance of Different Frameworks on the Training Corpus

| Feature Types | Classifier | Precision | Recall | Fmeasure |
|---|---|---|---|---|
| Entropy Based Features | AdaBoost | 0.80 | 0.83 | 0.81 |
| | Logistic Regression | 0.90 | 0.84 | 0.86 |
| | Random Forest | 0.93 | 0.83 | **0.87** |
| | Support Vector Machine | 0.91 | 0.88 | **0.89** |
| TF-IDF Based Features | AdaBoost | 0.77 | 0.83 | 0.79 |
| | Logistic Regression | 0.87 | 0.84 | 0.85 |
| | Random Forest | 0.88 | 0.82 | 0.84 |
| | Support Vector Machine | 0.94 | 0.81 | **0.87** |
| Doc2Vec Based Features | AdaBoost | 0.89 | 0.85 | **0.86** |
| | Logistic Regression | 0.78 | 0.82 | 0.80 |
| | Random Forest | 0.85 | 0.86 | **0.85** |
| | Support Vector Machine | 0.82 | 0.77 | 0.79 |

## 3.2. Analysis of Results

Initially, we have implemented four classifiers using three different feature engineering schemes individually on the training corpus. The performance of these frameworks are reported in Table 2 in terms of Precision, Recall and Fmeasure. These results are useful to analyze the performance of different proposed frameworks. Thereafter, the top five frameworks from Table 2 in terms of fmeasure have been selected and subsequently implemented on the given test corpus. Eventually the performance of these five frameworks on the test corpus are communicated as official results of our team.

It can be seen from Table 2 that SVM performs better than the other classifiers using Entropy based term weighting scheme following fmeasure. SVM outperforms the other classifiers for TF-IDF based term weighting scheme too in terms of fmeasure. For Doc2Vec based features, Ada Boost beats the other classifiers based on fmeasure. Table 2 shows that random forest classifier using both entropy based features and Doc2Vec based features cannot beat the other three frameworks, but they perform better than the rest of the methods in terms of fmeasure. Thus we have chosen these five frameworks following their performance on the training corpus as reported in Table 1 and have run them on the test corpus.

The results of the five runs on the test corpus in terms of precision, recall, fmeasure, $ERDE_5$

**Table 3**
Performance of Five Runs on the Test Corpus Following Different Evaluation Measures

| Runs | Precision | Recall | Fmeasure | $ERDE_5$ | $ERDE_{50}$ | $Latency_{TP}$ | Speed |
|---|---|---|---|---|---|---|---|
| Birmingham 0 (Entropy+SVM) | 0.584 | 0.526 | 0.554 | 0.068 | 0.054 | 2 | 0.996 |
| Birmingham 1 (Entropy+RF) | 0.644 | 0.309 | 0.418 | 0.097 | 0.074 | 8 | 0.973 |
| Birmingham 2 (TFIDF+SVM) | 0.757 | 0.349 | 0.477 | 0.085 | 0.07 | 4 | 0.988 |
| Birmingham 3 (Doc2Vec+AB) | 0.629 | 0.434 | 0.514 | 0.084 | 0.062 | 5 | 0.984 |
| Birmingham 4 (Doc2Vec+RF) | 0 | 0 | 0 | 0.105 | 0.105 | - | - |

[33], $ERDE_{50}$ [33], $Latency_{TP}$ [3] and speed [3] , are reported in Table 3. It can be seen from this table that the precision of the Birmingham 2 framework is better than the precision of the other Birmingham frameworks and it is the best precision score among the precision scores of all 55 submissions in Task 2 of eRisk2021 challenge. Birmingham 0 framework outperforms the other Birmingham frameworks in terms of recall, f-measure, $ERDE_5$, $ERDE_{50}$, $Latency_{TP}$ and speed, but its performance does not belong to the top three scores of all the submissions of Task 2 in eRisk 2021 challenge.

The latency and speed are two important measures for early prediction of risk of a disease over internet [3]. The latency and speed of the proposed frameworks are reasonably well, but many teams perform better than us in terms of speed and latency. In future we should look into this issue to improve the speed and latency. The bag of words model and SVM classifier generally perform well for classical text classification tasks. However, this framework can not achieve the semantic interpretation of the words in free texts. Hence it does not perform very well for text data over internet as these data sets contain irregular texts of diverse meaning. The deep learning based model can get rid of this situation and therefore we used the Doc2Vec based model to capture the semantic interpretation of the online texts. It can be observed from Table 3 that Doc2Vec based model could not perform well on the test corpus like the bag of words model. The deep learning based models works well when trained on large corpora [28]. The Doc2Vec based model performs poorly as we have trained it on the training corpus of the eRisk 2021 self-harm identification task, which is reasonably small.

## 4. Conclusion

The Task 2 of eRisk 2021 challenge aims to develop text mining tools for early prediction of risk of self harm over social media. Various text mining frameworks have been presented here using different types of features from the free text to accomplish this task. We have examined the performance of both bag of words features and Doc2Vec based features using different classifiers to identify signs of self harm. It has been observed from the experimental results that the conventional bag of words model performs better than the Doc2Vec model on the test corpus. Note that we have developed the Doc2Vec based embeddings based on the given training corpus, which has reasonably low number of documents in compare to the other pretrained deep learning based word embeddings e.g., Glove, which were trained on huge text collections. As a result the Doc2Vec model cannot properly represent the semantic interpretations of the given documents and hence its performance is not as good as the classical bag of words model. In future we can develop a large training corpus by collecting data from Reddit and similar forums for early risk prediction of self harm. Furthermore, we plan to develop some pretrained transformer based embeddings for depression and other mental disorders by collecting documents over social media, Wikipedia and publications to further improve the performance.

## Acknowledgments

This work was directly supported by the MRC Heath Data Research UK (HDRUK/CFC/01), an initiative funded by UK Research and Innovation, Department of Health and Social Care (England) and the devolved administrations, and leading medical research charities. The views expressed in this publication are those of the authors and not necessarily those of the NHS, the National Institute for Health Research, the Medical Research Council or the Department of Health. Georgios V. Gkoutos also acknowledges support from the NIHR Birmingham ECMC, NIHR Birmingham SRMRC, Nanocommons H2020-EU (731032) and the NIHR Birmingham Biomedical Research Centre.

## References

[1] M. De Choudhury, M. Gamon, S. Counts, E. Horvitz, Predicting depression via social media., ICWSM 13 (2013) 1–10.

[2] M. De Choudhury, S. Counts, E. Horvitz, Social media as a measurement tool of depression in populations, in: Proceedings of the 5th Annual ACM Web Science Conference, ACM, 2013, pp. 47–56.

[3] D. E. Losada, P. Martin-Rodilla, F. Crestani, J. Parapar, Overview of erisk 2021: Early risk prediction on the internet, in: Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2021.

[4] J. S. Obeid, J. Dahne, S. Christensen, S. Howard, T. Crawford, L. J. Frey, T. Stecker, B. E. Bunnell, Identifying and predicting intentional self-harm in electronic health record clinical notes: Deep learning approach, Journal of Medical Informatics Research 8 (2020) e17784.

[5] J. Robinson, K. Witt, M. Lamblin, M. J. Spittal, G. Carter, K. Verspoor, A. Page, G. Rajaram, V. Rozova, N. Hill, et al., Development of a self-harm monitoring system for victoria, International Journal of Environmental Research and Public Health 17 (2020) 9385.

[6] H. Bergen, K. Hawton, K. Waters, J. Ness, J. Cooper, S. Steeg, N. Kapur, Premature death after self-harm: a multicentre cohort study, The Lancet 380 (2012) 1568–1574.

[7] B. Mars, J. Heron, C. Crane, K. Hawton, G. Lewis, J. Macleod, K. Tilling, D. Gunnell, Clinical and social outcomes of adolescent self harm: population based birth cohort study, British Medical Journal 349 (2014).

[8] M.-H. Metzger, N. Tvardik, Q. Gicquel, C. Bouvry, E. Poulet, V. Potinet-Pagliaroli, Use of emergency department electronic medical records for automated epidemiological surveil-

lance of suicide attempts: a french pilot study, International Journal of Methods in Psychiatric Research 26 (2017) e1522.

[9] D. E. Losada, F. Crestani, J. Parapar, Overview of erisk 2019: Early risk prediction on the internet, in: Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2019, pp. 340–357.

[10] D. E. Losada, F. Crestani, J. Parapar, Overview of erisk 2020: Early risk prediction on the internet, in: Proceedings of the International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2020, pp. 272–287.

[11] S. G. Burdisso, M. Errecalde, M. Montes-y Gómez, A text classification framework for simple and effective early depression detection over social media streams, Expert Systems with Applications 133 (2019) 182–197.

[12] S. G. Burdisso, M. Errecalde, M. Montes-y Gómez, Unsl at erisk 2019: a unified approach for anorexia, self-harm and depression detection in social media., in: CLEF (Working Notes), 2019.

[13] N. Naderi, J. Gobeill, D. Teodoro, E. Pasche, P. Ruch, A baseline approach for early detection of signs of anorexia and self-harm in reddit posts., in: CLEF (Working Notes), 2019.

[14] R. Martínez-Castaño, A. Htait, L. Azzopardi, Y. Moshfeghi, Early risk detection of self-harm and depression severity using bert-based transformers: ilab at clef erisk 2020, in: CLEF (Working Notes), 2020.

[15] E. C. Ageitos, J. Martínez-Romo, L. Araujo, Nlp-uned at erisk 2020: Self-harm early risk detection with sentiment analysis and linguistic features., in: CLEF (Working Notes), 2020.

[16] C. D. Manning, P. Raghavan, H. Schutze, Introduction to Information Retrieval, Cambridge University Press, New York, 2008.

[17] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of International Conference on Machine Learning, 2014, pp. 1188–1196.

[18] N. B. Unnam, P. K. Reddy, A document representation framework with interpretable features using pre-trained word embeddings, International Journal of Data Science and Analytics 10 (2020) 49–64.

[19] G. Salton, M. J. McGill, Introduction to Modern Information Retrieval, McGraw Hill, 1983.

[20] A. Selamat, S. Omatu, Web page feature selection and classification using neural networks, Information Sciences 158 (2004) 69–88.

[21] T. Sabbah, A. Selamat, M. H. Selamat, F. S. Al-Anzi, E. H. Viedma, O. Krejcar, H. Fujita, Modified frequency-based term weighting schemes for text classification, Applied Soft Computing 58 (2017) 193–206.

[22] Y. Freund, R. Schapire, N. Abe, A short introduction to boosting, Journal-Japanese Society For Artificial Intelligence 14 (1999) 1612.

[23] A. Genkin, D. D. Lewis, D. Madigan, Large-scale bayesian logistic regression for text categorization, Technometrics 49 (2007) 291–304.

[24] B. Xu, X. Guo, Y. Ye, J. Cheng, An improved random forest classifier for text categorization., JCP 7 (2012) 2913–2920.

[25] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, Journal of machine learning research 2 (2001) 45–66.

[26] T. Basu, C. Murthy, A supervised term selection technique for effective text categorization, International Journal of Machine Learning and Cybernetics 7 (2016) 877–892.

[27] S. Paul, S. K. Jandhyala, T. Basu, Early detection of signs of anorexia and depression over social media using effective machine learning frameworks., in: In Proceedings of CLEF Working Notes, 2018.

[28] T. Basu, S. Goldsworthy, G. V. Gkoutos, A sentence classification framework to identify geometric errors in radiation therapy from relevant literature, Information 12 (2021) 139.

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[30] H. Aman, S. Amasaki, T. Yokogawa, M. Kawahara, A doc2vec-based assessment of comments and its application to change-prone method analysis, in: 2018 25th Asia Pacific Software Engineering Conference, IEEE, 2018, pp. 643–647.

[31] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: A library for large linear classification, Journal of machine learning research 9 (2008) 1871–1874.

[32] R. E. Schapire, Y. Singer, A. Singhal, Boosting and rocchio applied to text filtering, in: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1998, pp. 215–223.

[33] D. E. Losada, F. Crestani, A test collection for research on depression and language use, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2016, pp. 28–39.

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, Journal of machine learning research 12 (2011) 2825–2830.