

Early Risk Prediction by means of DeepLearning*

Pablo Raez Garcia Retamero and Isabel Segura Bedmar

¹ Universidad Carlos III de Madrid, praez@pa.uc3m.es

² Universidad Carlos III de Madrid, isegura@inf.uc3m.es

Abstract. This work presents our five approaches to early risk detection of anorexia on social media in CLEF eRisk 2019. Our models make use of different kinds of deep neural networks to classify the users in a danger situation. We show the effectiveness of our models by using the validation and test datasets. The best model obtains a F1 score of 0.57 over the objective class in the validation and a 0.20 over the test.

Keywords: Deep Learning · Natural Language Processing · Risk Prediction.

1 Introduction

Anorexia is an eating disorder which presents symptoms such as fear of gaining weight or a distorted and delirious perception of the own body. This disease is often associated with severe psychological alterations that cause changes in the emotional behaviour. These psychological alterations are discernible in the behaviour of the affected and are usually reflected in social media as posts and comments. Currently several anorexia detection methods exist [11, 2, 19, 14, 18, 13], which are mainly based in behavioural analysis. Anorexia symptoms are usually very diverse and probably hidden by the subjects of study, which makes it harder to make a decision, delaying the diagnoses.

Much research has been carried out to early detect these symptoms in social media in an automatic way. Even being a well known problem, anorexia is still hard to diagnose, due to it having wide variety of symptoms as well as the long periods needed for them to show up, as in the amenorrhea case [5]. Because getting to diagnose the patients is an arduous task, patients will receive treatment in later stages of anorexia. This, in turn, will make the therapy longer and more expensive than if the problem was promptly diagnosed. The automatic detection, with the highest possible accuracy, of anorexia in its early stages would mean great time savings as well as considerable patient health improvements who had been treated quickly.

* Supported by UC3M.

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CLEF 2019, 9-12 September 2019, Lugano, Switzerland.

Five different approaches were carried out in order to address this problem. These approaches are explained in further details in section 4. Both, the results obtained by the validation and testing dataset are included.

The paper is structured as follows. Section 2 gathers the state of the art of Natural Language Processing techniques applied to the risk prediction domain. Next, in section 3 the dataset and tools used are named. It is followed by section 4 where the methods as well as the neural architectures proposed are described. In section 5 the results obtained are shown. Finally in section 6 the conclusions and the future work are gathered.

2 State of the Art

This section gathers the main works related to early risk prediction on the internet. The usage of machine learning techniques in mental illness detection such as anorexia is quite recent. Even so, there is considerable bibliography on the matter [11, 2, 19, 14, 18, 13].

In [19], Deep Learning techniques have been applied to the problem of anorexia and depression detection for the CLEF eRisk 2018 tasks [9]. The authors approach the problem by turning it into a sentence classification one, where the sentences are classified as positive if they have been written by an ill user and negative otherwise. They make use of the TF-IDF algorithm to get the most representative words for each one of the classes. Then, the sentences are encoded by means of a Convolutional Neural Network (CNN). They managed to obtain F1 scores of 0.64 and 0.85 as well as ERDE5 of 8.78 and 11.40 in the depression and anorexia tasks, respectively.

Our first approach is quite similar to the one previously described, but we also make use of word or char embeddings in every model, as well as a fully connected layer after the CNN ones, which have been shown to improve the results of the classifier.

In [14] approach to the CLEF eRisk 2018 tasks, different machine learning techniques are presented, such as Linear Regression [12], Super Vector Machines [16], Ada Boost [15], Random Forests [1], and Recursive Neural Network (RNN) [17]. Texts are represented using different features such as Bag Of Words (BOW) and Unified Medical Language System (UMLS). Experiments show that the best results are obtained by BOW and using the classifiers Ada Boost and the Random Forests. They managed to obtain F1 scores of 0.58 and 0.67 as well as ERDE5 of 9.81 and 12.17 in the depression and anorexia tasks, respectively.

In [18] approach to the CLEF eRisk 2017 task [8], several combinations of user-level linguistic metadata, BoW [21], neural word embeddings [3], and CNN [7] are used. Obtaining an F1 value of 0.48 and an ERDE5 of 12.73 on the depression task.

There have been some interesting approaches not so heavily focused into machine and deep learning techniques such as the one described in [13], which focuses into Author Profiling (AP). It consists in analysing texts to predict

general or demographic attributes of authors such as: gender, age, personality, native language, and political orientation, among others.

3 Materials

This section gathers the materials used.

3.1 Dataset

The dataset for this task has the same format as the one described in [10]. The collection provided, for training and validation, is composed by 152 subjects, of them 20 are anorexic and 132 are not. The texts from these subjects are formed by a total of 253,341 posts and comments, of which 24,874 come from ill subjects and 228,467 are from healthy people. As it can be seen, the training set is very unbalanced, which in turn makes the whole task harder to perform.

For every different subject, we get all their writings with several information fields, being them the title of the post (sometimes blank), as well as the date and time. It also contains info about the platform where the post was made, may it be reddit or other, and the posted text itself.

The test dataset is hosted as a server that iteratively yields user writings to the participating teams. These iterations go across time to get the writings of each user in a more real-world-like scenario. It will only give back the writings when all runs of a timestep for a team are sent. This dataset counts with 2000 timestep for over 800 users. Being them "id", "nick", "redditor", "title", "content", and "date". The "nick" is used as the subject id, and the "title", "content" and "date" ones are used as their homonyms in the training dataset. "Redditor" and "id" do not relate with any of the training dataset and finally number indicates the iteration on the test dataset, which is used for validation purposes.

3.2 Tools

Google Colab was used to run the experiments. It consists of a machine with an Intel(R) Xeon(R) CPU @ 2.20GHz as a CPU and its equipped with 12Gb of RAM. The most interesting part of it for us is the GPU they provide, being it a Tesla K80 GPU with 12Gb of memory as well.

The experiments were developed using python, and its libraries Keras and Tensorflow for DL models. Some other libraries were used such as Pandas or NumPy for the processing of the data.

4 Method

In this section the method followed for the development of the approaches is explained. This method includes all the pre and post processing of the data.

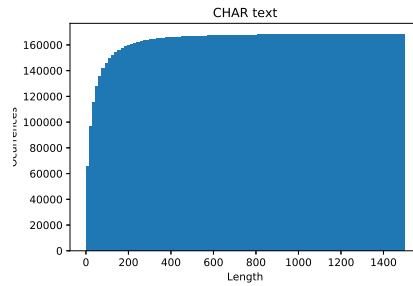
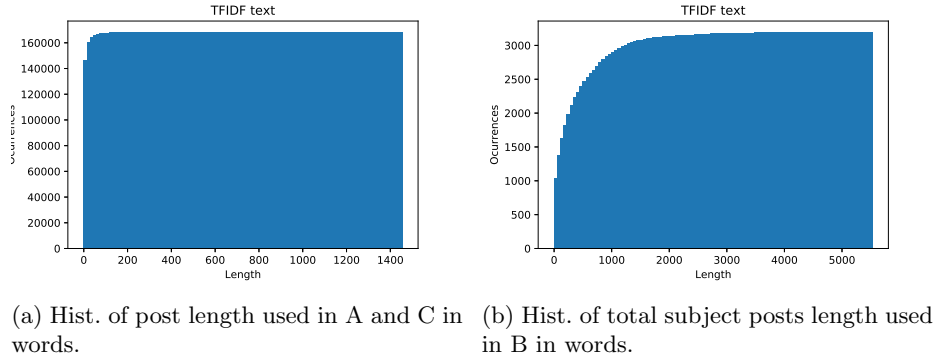


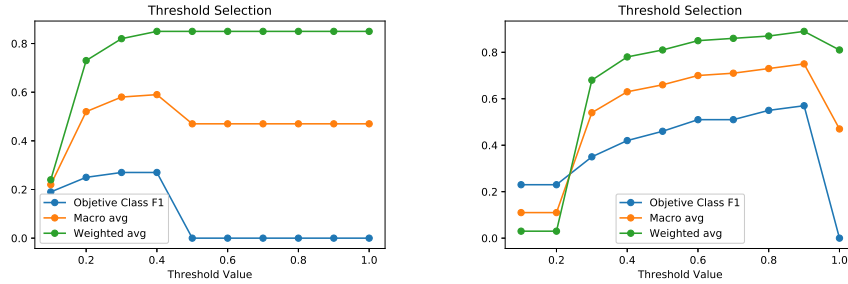
Fig. 1: Histograms of length of posts of models A and C, B, and D and E.

Different types of neural networks such as RNN and CNN have been used to generate deep learning models, which are further explained below.

As a preprocessing step, all texts are cleaned by removing stop words, numbers, punctuation and words with less than three characters. Then a TF-IDF algorithm is used in order to lower the volume of words while retaining the most representative ones.

For the models A, B and C, the posts are tokenized and cropped or padded to a fixed length of 50 words per post in models A and C. This padding and cropping takes place because the input for the neural networks must have a fixed shape. The reason why longer texts are cropped is because too much padding will add too much noise to the networks. This is because than most texts have less than 50 words, as shown in figure 1a. The selected length of the B model is 350. Contrary to the one selected in the previous models, this length is chosen due to the prohibitive size of the network past it. The ideal value would have been 1000, as can be seen in figure 1b.

For the models E and D instead of tokenizing the texts and fixing them to a certain length, another preprocessing step is added, based in splitting the words into characters. This operation is needed in order to make use of char embed-



(a) F1 scores obtained by A model depending on threshold value (b) F1 scores obtained by C model depending on threshold value

Fig. 2: F1 variations with variable threshold experiments

dings, which have shown themselves useful in NLP tasks [20]. Char embeddings has its advantages against word embeddings: they do not face problems when processing unseen words as every word is formed with characters. Another characteristic advantage is the robustness against misspelled words. Furthermore, char embeddings are usually low dimensional ones, which in turn improves the speed of the models. Then each text is fixed to a length of 400 characters. The length was picked by hand and it was done regarding the fig 1c in the same way as for models A, B and C. Finally the characters were fed into the different neural networks.

Finally, and after the processing performed by the different models, the output of the networks is compared with a threshold to determine if it was a risk situation or not. This threshold was obtained empirically for each model by subdividing the results to several tests in which the threshold value iterated in the range of 0.1 and 0.9. Then, the threshold with the highest F1 of the active class was selected. The thresholds are shown in table 1. An illustrative example of this process can be seen in figure 2 where the evaluation of A and C thresholds is shown.

Table 1: Best found model thresholds.

Models	A	B	C	D	E
Threshold	0.4	0.1	0.9	0.3	0.6

Several experiments were performed to find out the best hyper-parameter configuration for each one of the models, which can be found in table 3. The tuned hyperparameters regarding the model can be found in detail in table 2

Some of the hyper-parameters checked were regarding the model themselves, such as *load_emb*, *emb_size*, *trainable_emb*, *cnn_size*, *rnn_size*, *dropout*, *dnn_size*, and *batch_size*. Some others were specific of the type of networks used; in the


Table 2: Hyperparameters tuned in the experiments.

Hyperparameter	Type	Explanation
load_emb	Boolean	Determines if the model will use pre-trained embeddings
emb_size	Numerical	Defines the dimension of the embedding vector
trainable_emb	Boolean	Determines if the embeddings are further trainable or not
cnn_size	Vector	Defines not only the number of kernels of the CNN, but also the number of stacked CNN layers the model will have
cnn_filter	Vector	Defines the size of the kernel used, as well as the number of them
rnn_size	Vector	Defines the number of stacked RNN layers as well as their dimensions
cell_type	Categorical	Determines the RNN cell used will be a LSTM or a GRU one
bidirectional	Boolean	Defines if a bidirectional RNN is used or not
attention	Boolean	Determines if the output of the RNN goes through an attention mechanism
dropout	Numerical	Defines the dropout the network will be using in training phase
dnn_size	Vector	Defines the number of stacked fully connected layers as well as their dimensions
batch_size	Numerical	Defines the number of instances processed in the same batch

CNN was `cnn.filter`, which determines the size of the kernel used, and in the RNN we can find `cell.type`, determining the type of the cell used, being it GRU or LSTM, `bidirectional` that indicates if the layers were bidirectional ones, and `attention` which, as its own name depicts, determines if an attention mechanism was used or not.

4.1 A model

This model is a simple first approach to classify the different records independently. The posts are taken as if they were independent, and they are labelled to 0 or 1 taking into account if the user who wrote them was control class or positive class patient.

This model gets as input the different texts, which then will undergo a Word Embedding layer, whose output is fed to a one-dimensional CNN. Finally, the output of the former layer is fed into a fully connected layer just before the output one (see .

4.2 B model

This model similar approach to the the previous one. But in this case, instead of taking the texts as independent bits of information, all of the texts of the same user are processed together. This way, the input to the net is all the tokenized text a user has ever posted and the objective value is if the subject is in risk of suffering anorexia or not.

This model gets the text input which, in the same way as in the previous model, undergo a Word Embedding layer, whose output is in the same way fed

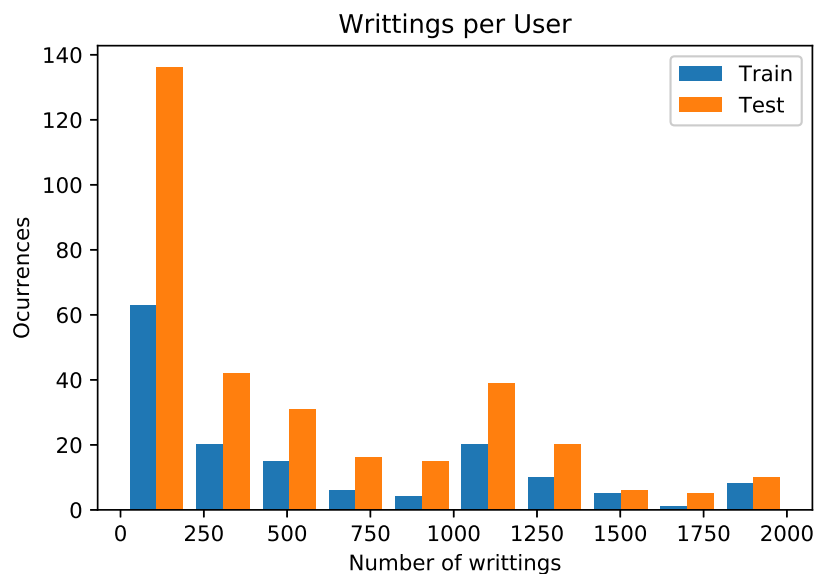


Fig. 3: Number of writings per user training and validation datasets

to a RNN layer. The result is then fed to a fully connected layer which is placed just before the output one (see Img 4b).

4.3 C model

This model is a more sophisticated one in the sense that it uses previous A models in order to generate what we call "writing embeddings" by means of transfer learning [6]. Then they are fed to a RNN layer, which allows us to process varying number of texts. This is crucial due to the dataset having very variable number of texts per user as can be seen in fig 3.

It is composed of the whole best A model without the two last layers. Those outputs are used as "writing embeddings" which represent the different texts in just a 32 dimension vector. Then, the "writing embeddings" are fed into the RNN layers, whose output is then passed through a fully connected layer before the output layer (see Img 5a).

4.4 D model

This model follows the same idea as the A, which is to classify the different texts independently. But it differs from the previous one in the fact that it does not use word embeddings, but char embeddings instead.

This model gets as input the different chars from every post, which then will undergo a Char Embedding layer, which mainly differs from the word embedding

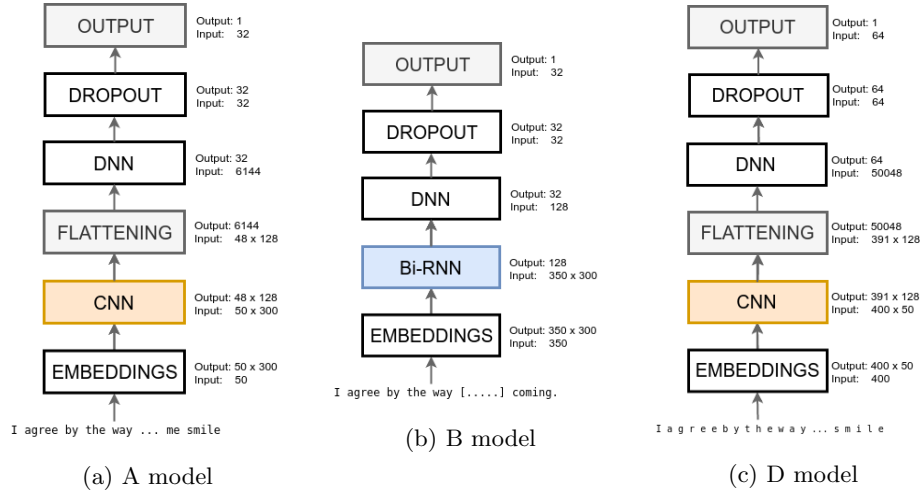


Fig. 4: Structure of models A, B and D.

one in the dimensions of the vector which is way shorter, as well as in the vocabulary which is, as well, way smaller. The output of this layer is then fed into a one-dimensional CNN in the same way as the A model. Finally, the output of the CNN is fed into a fully connected layer, and then the output one (see Img 4c).

4.5 E model

This model follows the same idea as the C, which is to use a pre-trained model to generate "writing embeddings". But it differs from the previous one in the fact that instead of using a pre-trained model on word embeddings, it uses a char embeddings pre-trained model. This model also takes advantage on the RNN layers that allow it to process the users no matter the number of texts they individually have, which, as aforementioned, is really disperse (see Fig 3).

This model makes use of the best D model weights, but without the two last layers. The outputs resulting of the processing with the cropped D model, which are given in the form of a 64 dimension vector, are fed into the RNN layers. Finally, likewise the previous models, the output of the former layer is fed into a fully connected network, and then it goes under the output one (see Img 5b).

5 Results

In this section, the results obtained by the five different approaches are shown. We divide this section in the validation results and the test results. The evaluation has taken place by means of the test server presented in section 3. We also include the best results obtained in the challenge.

Table 3: Best found model configuration.

Models	Emb Size	CNN cells	Kernel Size	RNN	RNN Type	Bidirectional	FNN	Batch Size
A	300	128	3	None	None	None	32	1024
B	300	None	None	64	GRU	True	32	32
C	None	None	None	64	LSTM	True	32	1
D	50	128	10	None	None	None	64	1024
E	None	None	None	64	GRU	False	32	1

Table 4: F1 validation scores achieved by the models.

Model	A	B	C	D	E
F1 (class 1)	0.27	0.12	0.57	0.26	0.23
Macro F1	0.59	0.52	0.75	0.54	0.57
Weighted	0.85	0.81	0.89	0.77	0.83

The common measure of performance in terms of precision and recall is the F1-score [4]. This metric is the harmonic mean of the precision and recall. As we are mostly concerned about the performance over the positive class, only the F1 of that class is shown in the validation results. We also add the Macro F1 due to it being a good measure of the performance with unbalanced classes, where the most important is the least represented one. Finally we add the weighted F1 as a comparison.

The best results of each model can be seen in the table 4. The best results are in bold. These metrics are very limited in comparison with the ones provided by the challenge organisers. Still the validation metrics provided are promising, specially the ones obtained by the C approach. Still further work must be performed in order to improve the overall results.

The results obtained in the official evaluation are shown in table 5. The best results obtained for each metric are also shown in the aforementioned table.

6 Conclusions and Future Work

Five different approaches to the CLEF eRisk 2019 task 1 have been described. All the approaches make use of some kind of neural networks and two of them benefit from concepts such as transfer learning. Several hyper-parameters of those models were finely-tuned in order to achieve the better performance possible. Although our official results are very low, we can conclude that our models provide promising results for the early detection of anorexia in social media, obtaining an F1 score up to a 0.57 in the positive class. Not so good results were obtained in the test experimentation, F1-wise, even so, for ERDE5 and ERDE50, results close to the best ones were obtained.

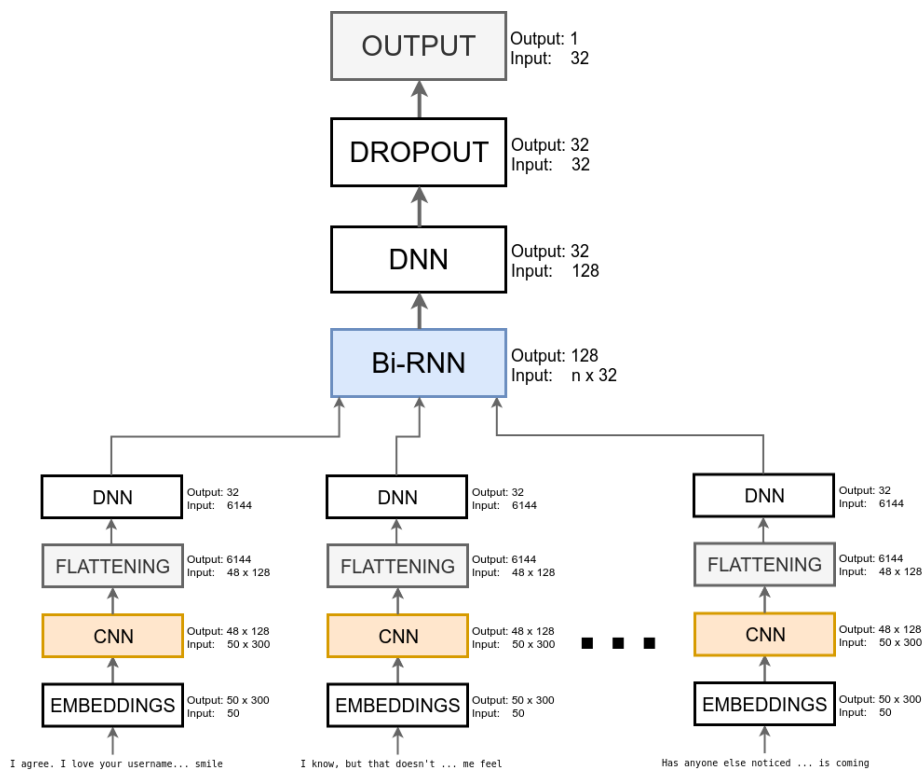
Still, further work is needed. We would like to feed the different approaches with more kinds of embeddings such as concept embeddings, as well as to put to test the usage of word embeddings and char embeddings in the same model.

Table 5: F1 official scores achieved by the models.

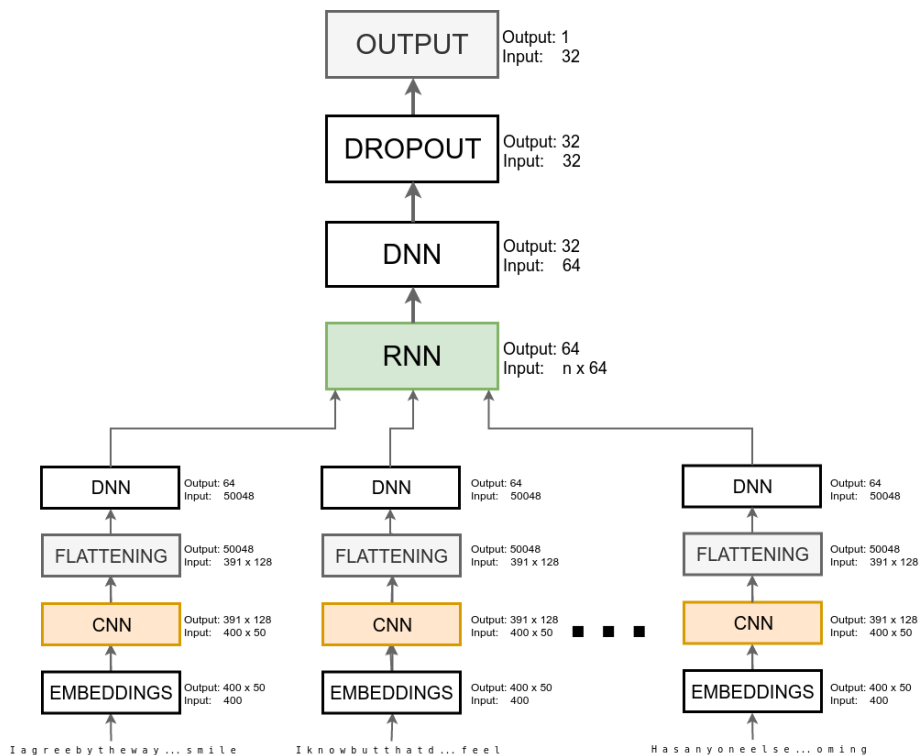
Model	A	B	C	D	E	Best
F1	0.20	0.15	0.13	0.16	0.16	0.71
ERDE5	0.07	0.11	0.13	0.09	0.10	0.06
ERDE50	0.07	0.08	0.12	0.07	0.08	0.03
Latency - Weighted F1	0.20	0.15	0.09	0.16	0.16	0.69

Acknowledgments

This work was supported by the Research Program of the Ministry of Economy and Competitiveness - Government of Spain, (DeepEMR project TIN2017-87548-C2-1-R).



(a) C model



(b) E model

Fig. 5: Structure of models C and E.

References

1. Breiman, L.: Random forests. *Machine learning* **45**(1), 5–32 (2001)
2. Coppersmith, G., Dredze, M., Harman, C.: Quantifying mental health signals in twitter. In: *Proceedings of the workshop on computational linguistics and clinical psychology: From linguistic signal to clinical reality*. pp. 51–60 (2014)
3. Goldberg, Y., Levy, O.: word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014)
4. Goutte, C., Gaussier, E.: A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: *European Conference on Information Retrieval*. pp. 345–359. Springer (2005)
5. Gutiérrez-Barquín, I.E.: Alteraciones menstruales y anorexia nerviosa. *Trastornos de la conducta alimentaria* (3), 277–284 (2006)
6. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2014)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. pp. 1097–1105 (2012)
8. Losada, D.E., Crestani, F., Parapar, J.: erisk 2017: Clef lab on early risk prediction on the internet: experimental foundations. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. pp. 346–360. Springer (2017)
9. Losada, D.E., Crestani, F., Parapar, J.: Overview of erisk: Early risk prediction on the internet. In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. pp. 343–361. Springer (2018)
10. Losada, D.E., Crestani, F., Parapar, J.: Overview of eRisk 2019: Early Risk Prediction on the Internet. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 10th International Conference of the CLEF Association, CLEF 2019*. Springer International Publishing, Lugano, Switzerland (2019)
11. Mohr, D.C., Zhang, M., Schueller, S.M.: Personal sensing: Understanding mental health using ubiquitous sensors and machine learning. *Annual review of clinical psychology* **13**, 23–47 (2017)
12. Montgomery, D.C., Peck, E.A., Vining, G.G.: *Introduction to linear regression analysis*, vol. 821. John Wiley & Sons (2012)
13. Ortega-Mendoza, R.M., López-Monroy, A.P., Franco-Arcega, A., Montes-y Gómez, M.: Peimex at erisk2018: Emphasizing personal information for depression and anorexia detection
14. Paul, S., Kalyani, J.S., Basu, T.: Early detection of signs of anorexia and depression over social media using effective machine learning frameworks
15. Schapire, R.E.: Explaining adaboost. In: *Empirical inference*, pp. 37–52. Springer (2013)
16. Scholkopf, B., Smola, A.J.: *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press (2001)
17. Socher, R., Huang, E.H., Pennin, J., Manning, C.D., Ng, A.Y.: Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In: *Advances in neural information processing systems*. pp. 801–809 (2011)
18. Trotzek, M., Koitka, S., Friedrich, C.M.: Word embeddings and linguistic metadata at the clef 2018 tasks for early detection of depression and anorexia

19. Wang, Y.T., Huang, H.H., Chen¹², H.H.: A neural network approach to early risk detection of depression and anorexia on social media text
20. Zhang, X., LeCun, Y.: Text understanding from scratch. arXiv preprint arXiv:1502.01710 (2015)
21. Zhang, Y., Jin, R., Zhou, Z.H.: Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics* **1**(1-4), 43–52 (2010)