Enhanced Word Embeddings for Anorexia Nervosa Detection on Social Media

Diana Ramírez-Cifuentes[1(✉)], Christine Largeron[2], Julien Tissier[2], Ana Freire[1], and Ricardo Baeza-Yates[1]

[1] Universitat Pompeu Fabra, Carrer de Tanger, 122-140, 08018 Barcelona, Spain
{diana.ramirez,ana.freire,ricardo.baeza}@upf.edu
[2] Univ Lyon, UJM-Saint-Etienne, CNRS, Laboratoire Hubert Curien, UMR 5516, 42023 Saint-Etienne, France
{julien.tissier,christine.largeron}@univ-st-etienne.fr

Abstract. Anorexia Nervosa (AN) is a serious mental disorder that has been proved to be traceable on social media through the analysis of users' written posts. Here we present an approach to generate word embeddings enhanced for a classification task dedicated to the detection of Reddit users with AN. Our method extends *Word2vec*'s objective function in order to put closer domain-specific and semantically related words. The approach is evaluated through the calculation of an average similarity measure, and via the usage of the embeddings generated as features for the AN screening task. The results show that our method outperforms the usage of fine-tuned pre-learned word embeddings, related methods dedicated to generate domain adapted embeddings, as well as representations learned on the training set using *Word2vec*. This method can potentially be applied and evaluated on similar tasks that can be formalized as document categorization problems. Regarding our use case, we believe that this approach can contribute to the development of proper automated detection tools to alert and assist clinicians.

Keywords: Social media · Eating disorders · Word embeddings · Anorexia Nervosa · Representation learning

1 Introduction

We present models to identify users with AN based on the texts they post on social media. Word embeddings previously learned in a large corpus, have provided good results on predictive tasks [3]. However, in the case of writings generated by users living with a mental disorder such as AN, we observe specific vocabulary exclusively related with the topic. Terms such as: "*cw*", used to refer to the current weight of a person, or "*ow*" referring to the objective weight,

This work was supported by the University of Lyon - IDEXLYON and the Spanish Ministry of Economy and Competitiveness under the Maria de Maeztu Units of Excellence Program (MDM-2015-0502).

M. R. Berthold et al. (Eds.): IDA 2020, LNCS 12080, pp. 404–417, 2020.
https://doi.org/10.1007/978-3-030-44584-3_32

are elements that are not easily found in large yet general collections extracted from Wikipedia, social media and news websites. Therefore, using pre-learned embeddings may not be the most suitable approach for the task.

We propose a method based on *Dict2vec* [15] to generate word embeddings enhanced for our task domain. The main contributions of our work are the following: (1) a method that modifies *Dict2vec* [15] in order to generate word embeddings enhanced for our classification task, this method has the power to be applied on similar tasks that can be formulated as document categorization problems; (2) different ways to improve the performance of the embeddings generated by our method corresponding to four embeddings variants; and (3) a set of experiments to evaluate the performance of our generated embeddings in comparison to pre-learned embeddings, and other domain adaptation methods.

2 Related Work

In previous work related to detection of mental disorders [8], documents were represented using bag of words (BoW) models, which involve representing words in terms of their frequencies. As these models do not consider contextual information or relations between the terms, other models have been proposed based on word embeddings [3]. These representations are generated considering the distributional hypothesis, which assumes that words appearing in similar contexts are related, and therefore should have close representations [11,13].

Embedding models allow words from a large corpus to be encoded as vectors in a high-dimensional space. The vectors are defined by taking into account the context in which the words appear in the corpus in such a way that two words having the same neighborhood should be close in the vector space.

Among the methods used for generating word embeddings we find *Word2vec* [11], which generates a vector for each word in the corpus considering it as an atomic entity. To build the embeddings, *Word2vec* defines two approaches: one known as continuous bag of words (CBOW) that uses the context to predict a target word; and another one called skip-gram, which uses a word to predict a target context. Another method is *fastText* [2], which takes into account the morphology of words, having each word represented as a bag of character n-grams for training. There is also *GloVe* [13], which proposes a weighted least squares model that does the training on global word-word co-occurrence counts.

In contrast to the previous methods, we can also mention recent methods like Embeddings from Language Models (ELMo) [14] and Bidirectional Encoder Representations from Transformers (BERT) [6] that generate representations which are aware of the context they are being used at. These approaches are useful for tasks where polysemic terms are relevant, and when there are enough sentences to learn these from the context. Regarding our use case, we observe that the vocabulary used by users with AN is very specific and contains almost no polysemic terms, which is why these methods are not addressed in our evaluation framework.

All the methods already mentioned are generally trained over large general purpose corpora. However, for certain domain specific classification tasks we have to work with small corpora. This is the case of mental disorders screening tasks given that the annotation phase is expensive, and requires the intervention of specialists. There are some methods that address this issue by either enhancing the embeddings learned over small corpora with external information, or adapting embeddings learned on large corpora to the task domain.

Among the enhancement methods we find Zhang's *et al.* [17] work. They made use of word embeddings learned in different health related domains to recognize symptoms in psychiatry. They designed approaches to combine data of the source and target to generate word embeddings, which are considered in our experimental results.

Kuang *et al.* [9] propose learning weights based on the words' relative importance for the classification task (predictive terms). This method proposes weighting words according to their χ^2 [12] statistics to represent the context. However, this method differs from ours as we generate our embeddings through a different approach, which takes into account the context terms, introduces new domain related vocabulary, considers the predictive terms to be equally important, and moves apart the vectors of terms that are not predictive for the main target class.

Faruqui *et al.* [7] present an alternative, known as a retrofitting method, which makes use of relational information from semantic lexicons to improve pre-built word vectors. The main disadvantage is that no external new terms representations can be introduced to the enhanced embeddings, and that despite related embeddings are put closer, the embeddings of terms that should not be related (task-wise) cannot be put apart from each other. In our experimental setup, this method is used to define a baseline and to enhance the embeddings generated through our approach.

Our proposal is based on *Dict2vec* [15], which is an extension of the *Word2vec* approach. *Dict2vec* uses the lexical dictionary definitions of words in order to enrich the semantics of the embeddings generated. This approach has proved to perform well on small corpora because in addition to the context defined by *Word2vec*, it introduces a (1) positive sampling, which moves closer the vector of words co-occurring in their mutual dictionary definitions, and a (2) controlled negative sampling which prevents to move apart the vectors of words that appear in the definition of others, as the authors assume that all the words in the definition of a term from a dictionary are semantically related to the word they define.

3 Method Proposed

Our method generates word embeddings enhanced for a classification task dedicated to the detection of users with AN over a small size corpus. In this context, users are represented by documents that contain their writings concatenated, and that are labeled as anorexic (positive) or control (negative) cases. These labels are known as the classes to predict for our task.

Our method is based on *Dict2vec*'s general idea [15]. We extend the *Word2vec* model with both a positive and a negative component, but our method differs from *Dict2vec* because both components are designed to learn vectors for a specific classification task. Within the word embeddings context, we assume that word-level n-grams' vectors, which are predictive for a class, should be placed close to each other given their relation with the class to be predicted. Therefore we first define sets of what we call *predictive pairs* for each class, and use them later for our learning approach.

3.1 Predictive Pairs Definition

Prior to learning our embeddings, we use χ^2 [12] to identify the predictive n-grams. This is a method commonly used for feature reduction, being capable to identify the most predictive features, in this case terms, for a classification task.

Based on the χ^2 scores distribution, we obtain the n terms with the highest scores (most predictive terms) for each of the classes to predict (positive and negative). Later, we identify the most predictive term for the positive class denoted as t_1 or *pivot term*. Depending on the class for which a term is predictive, two types of *predictive pairs* are defined, so that every time a predictive word is found, it will be put close or far from t_1. These predictive pair types are: (1) positive predictive pairs, where each predictive term for the positive class is paired with the term t_1 in order to get its vector representation closer to t_1; and (2) negative predictive pairs, where each term predictive for the negative class is also paired with t_1, but with the goal of putting it apart from t_1.

In order to define the positive predictive terms for our use case, we consider: the predictive terms defined by the χ^2 method, AN related vocabulary (domain-specific) and the k most similar words to t_1 obtained from pre-learned embeddings, according to the cosine similarity. Like this, information coming from external sources that are closely related with the task could be introduced to the training corpus. The terms that were not part of the corpus were appended to it, providing us an alternative to add new vocabulary of semantic significance to the task.

Regarding the negative predictive terms, no further elements are considered asides from the (χ^2) predictive terms of the negative class as for our use case and similar tasks, control cases do not seem to share a vocabulary strictly related to a given topic. In other words, and as observed for the anorexia detection use case, control users are characterized by their discussions on topics unrelated to anorexia.

For the χ^2 method, when having a binary task, the resulting predictive features are the same for both classes (positive and negative). Therefore, we have proceeded to get the top n most predictive terms based on the distribution of the χ^2 scores for all the terms. Later, we decided to take a look at the number of documents containing the selected n terms based on their class (anorexia or control). Given a term t, we calculated the number of documents belonging to the positive class (anorexia) containing t, denoted as PCC; and we also calculated the number of documents belonging to the negative class (control) containing t,

named as NCC. Then, for t we calculate the respective ratio of both counts in relation to the total amount of documents belonging to each class: total amount of positive documents (TPD) and total amount of negative documents (TND), obtaining like this a positive class count ratio (PCCR) and a negative class count ratio (NCCR).

For a term to be part of the set of positive predictive terms its PCCR value has to be higher than the NCCR, and the opposite applies for the terms that belong to the set of negative predictive pairs. The positive and negative class count ratios are defined in Eqs. 1a and 1b as:

$$PCCR(t) = \frac{PCC(t)}{TPD} \tag{1a}$$

$$NCCR(t) = \frac{NCC(t)}{TND} \tag{1b}$$

3.2 Learning Embeddings

Once the predictive pairs are defined, the objective function for a target term ω_t (Eq. 2) is defined by the addition of a positive sampling cost (Eq. 3) and a negative sampling cost (Eq. 4a) in addition to *Word2vec*'s usual target, context pair cost given by $\ell(\omega_t, \omega_c)$ where ℓ represents the logistic loss function, and v_t, and v_c are the vectors associated to ω_t and ω_c respectively.

$$J(\omega_t, \omega_c) = \ell(v_t, v_c) + J_{pos}(\omega_t) + J_{neg}(\omega_t) \tag{2}$$

Unlike *Dict2vec*, J_{pos} is computed for each target term where $P(\omega_t)$ is the set of all the words that form a positive predictive pair with the word ω_t, and v_t and v_i are the vectors associated to ω_t and ω_i respectively. β_P is a weight that defines the importance of the positive predictive pairs during the learning phase. Also, as an aspect that differs from *Dict2vec*, the cost given by the predictive pairs is normalized by the size of the predictive pairs set, $|P(\omega_t)|$, considering that all the terms from the predictive pairs set of ω_t are taken into account for the calculations, and therefore when t_1 is found, the impact of trying to move it closer to a big amount of terms is reduced, and it remains as a pivot element to which other predictive terms get close to:

$$J_{pos}(\omega_t) = \beta_P \sum_{\omega_i \epsilon P(\omega_t)} \frac{\ell(v_t \cdot v_i)}{|P(\omega_t)|} \tag{3}$$

On the negative sampling, we modify *Dict2vec*'s approach. We not only make sure that the vectors of the terms forming a positive predictive pair with ω_t are not put apart from it, but we also define a set of words that are predictive for the negative class and define a cost given by the negative predictive pairs. In this case, as explained before, the main goal is to put apart these terms from t_1, so this cost is added to the negative random sampling cost J_{n_r} (Eq. 4b), as detailed in Eq. 4a.

$$J_{neg}(\omega_t) = J_{n\_r}(\omega_t) + \beta_N \sum_{\omega_j \in N(\omega_t)} \frac{\ell(-v_t \cdot v_j)}{|N(\omega_t)|} \tag{4a}$$

$$J_{n\_r}(\omega_t) = \sum_{\substack{\omega_i \in F_(\omega_t) \\ \omega_i \notin P_(\omega_t)}} \ell(-v_t \cdot v_i) \tag{4b}$$

The negative sampling cost considers, as on *Word2vec*, a set $F(\omega_t)$ of k words selected randomly from the vocabulary. These words are put apart from ω_t as they are likely to not be semantically related. Considering *Dict2vec*'s approach, we make sure as well that any term belonging to the set of positive predictive pairs of ω_t ends up being put apart. In addition to this, we add another negative sampling cost which corresponds to the cost of putting apart from t_1 the most predictive terms from the negative class. In this case, $N(\omega_t)$ represents the set of all the words that form a negative predictive pair with the word ω_t. β_N is a weight to define the importance of the negative predictive pairs during the learning phase.

The global objective function (Eq. 5) is given by the sum of every pair's cost across the whole corpus:

$$J = \sum_{t=1}^{C} \sum_{c=-n}^{n} J(\omega_t, \omega_{t+c}) \tag{5}$$

where C is the corpora size, and n represents the size of the window.

3.3 Enhanced Embeddings Variations

Given a pre-learned embedding which associates for a word ω a pre-learned representation v_{pl}, and an enhanced embedding v obtained through our approach for ω with the same length m as v_{pl}, we generate variations of our embeddings based on existing enhancement methods. First, we denote the embeddings generated exclusively by our approach (predictive pairs) as *Variation 0*, v is an instance of the representation of ω for this variation.

For the next variations, we address ways to combine the vectors of pre-learned embeddings (*i.e.*, v_{pl}) with the ones of our enhanced embeddings (*i.e.*, v). For *Variation 1* we concatenate both representations $v_{pl} + v$, obtaining a *2m* dimensions vector [16]. *Variation 2* involves concatenating both representations and applying truncated SVD as a dimensionality reduction method to obtain a new representation given by $SVD(v_{pl} + v)$. *Variation 3* uses the values of the pre-learned vector v_{pl} as starting weights to generate a representation using our learning approach. This variation is inspired in a popular transfer learning method that was successfully applied on similar tasks [5]. For these variations (1–3) we take into account the intersection between the vocabularies of both embeddings types (pre-learned and *Variation 0*). Finally, *Variation 4* implies applying Faruqui's retrofitting method [7] over the embeddings of *Variation 0*.

4 Evaluation Framework

4.1 Data Set Description

We used a Reddit data set [10] that consists on posts of users labeled as anorexic and control cases. This data set was defined in the context of an early risk detection shared task, and the training and test sets were provided by the organizers of the eRisk task.[1] Table 1 provides a description of the training and testing data sets statistics. Given the incidence of Anorexia Nervosa, for both sets there is a reduced yet significant amount of AN cases compared to the control cases.

Table 1. Collection description as described on [10].

	Train		Test	
	Anorexia	Control	Anorexia	Control
Users count	20	132	41	279
Writings count	7,452	77,514	17,422	151,364
Avg. writings count	372.6	587.2	424.9	542.5
Avg. words per writing	41.2	20.9	35.7	20.9

4.2 Embeddings Generation

The training corpus used to generate the embeddings, named *anorexia corpus*, consisted on the concatenation of all the writings from all the training users. A set of stop-words were removed. This resulted on a training corpus with a size of 1,267,208 tokens and a vocabulary size of 87,197 tokens. In order to consider the bigrams defined by our predictive pairs, the words belonging to a bigram were paired and formatted as if they were a single term.

For the predictive pairs generation with χ^2, each user is an instance represented by a document composed by all the user's posts concatenated. χ^2 is applied over the train set considering the users classes (anorexic or control) as the possible categories for the documents. The process described in Sect. 3.1 is followed in order to obtain a list of 854 positive (anorexia) and 15 negative (control) predictive terms. Some of these terms can be seen on Table 2, which displays the top 15 most predictive terms for both classes. *Anorexia* itself resulted to be the term with the highest χ^2 score, denoted as t_1 in Sect. 3.

The anorexia domain related terms from [1] were added as the topic related vocabulary, and the top 20 words with the highest similarity to *anorexia* coming from a set of pre-learned embeddings from *GloVe* [13] were also paired to it to define the predictive pairs sets. The GloVe's pre-learned vectors considered are the 100 dimensions representations learned over 2B tweets with 27B tokens, and with 1.2M vocabulary terms.

[1] eRisk task: https://early.irlab.org/2018/index.html.

Table 2. List of some of the most predictive terms for each class.

Positive Terms (Anorexia class)			Negative terms (Control class)		
anorexia	diagnosed	binges	war	sky	song
anorexic	macros	calories don't	bro	plot	master
meal plan	cal	relapsed	Trump	game	Russian
underweight	weight gain	restriction	players	Earth	video
eating disorder(s)	anorexia nervosa	caffeine	gold	America	trailer

The term *anorexia* was paired to 901 unique terms and, likewise, each of these terms was paired to *anorexia*. The same approach was followed for the negative predictive terms (15), which were also paired with *anorexia*. An instance of a positive predictive pair is *(anorexia, underweight)*, whereas an instance of a negative predictive pair is *(anorexia, game)*. For learning the embeddings through our approach, and as it extends *Word2vec*, we used as parameters a window size of 5, the number of random negative pairs chosen for negative sampling was 5, and we trained with one thread/worker and 5 epochs.

4.3 Evaluation Based on the Average Cosine Similarity

This evaluation is done over the embeddings generated through *Variation 0* over the anorexia corpus. It averages the cosine similarities (sim) between t_1 and all the terms that were defined either as its p positive predictive pairs, obtaining a positive score denoted as PS on Eq. 6a; or as its n negative predictive pairs, with a negative score denoted as NS on Eq. 6b. On these equations v_a represents the vector of the term *anorexia*; v_{PPT_i} represents the vector of the positive predictive term (PPT) i belonging to the set of positive predictive pairs of *anorexia* of size p; and v_{NPT_i} represents the vector of the negative predictive term (NPT) i belonging to the set of negative predictive pairs of *anorexia* of size n:

$$PS(a) = \frac{\sum_{i=1}^{p} sim(v_a, v_{PPT_i})}{p} \tag{6a}$$

$$NS(a) = \frac{\sum_{i=1}^{n} sim(v_a, v_{NPT_i})}{n} \tag{6b}$$

We designed our experiments using PS and NS in order to analyze three main aspects: (1) we verify that through the application of our method, the predictive terms for the positive class are closer to the pivot term representation, and that the predictive terms for the negative class were moved away from it; (2) we evaluate the impact of using different values of the parameters β_P and β_N to obtain the best representations where PS has the highest possible value, keeping NS as low as possible; and (3) we compare our generation method with *Word2vec* as baseline since this is the case for which our predictive pairs would not be considered ($\beta_P = 0$ and $\beta_N = 0$). We expect for our embeddings to obtain higher values for PS and lower values for NS in comparison to the baseline.

Results. Table 3 shows first the values for PS and NS obtained by what we consider our baseline, *Word2vec* ($\beta_P = 0$ and $\beta_N = 0$), and then the values obtained by embeddings models generated using our approach (*Variation 0*), with different yet equivalent values given to the parameters β_P and β_N, as they proved to provide the best results for PS and PN. We also evaluated individually the effects of varying exclusively the values for β_P, leaving $\beta_N = 0$, and then the effects of varying only the values of β_N, with $\beta_P = 0$. On the last row of the table we show a model corresponding to the combination of the parameters with the best individual performance ($\beta_P = 75$ and $\beta_N = 25$).

After applying our approach the value of PS becomes greater than NS for most of our generated models, meaning that we were able to obtain a representation where the positive predictive terms are closer to the pivot term *anorexia*, and the negative predictive terms are more apart from it. Then, we can also observe that the averages change significantly depending on the values of the parameters β_P and β_N, and for this case the best results according to PS are obtained when $\beta_P = 50$ and $\beta_N = 50$. Finally, when we compare our scores with *Word2vec*, we can observe that after applying our method, we can obtain representations where the values of PS and NS are respectively higher and lower than the ones obtained by the baseline model.

Table 3. Positive Scores (PS) and Negative Scores (NS) for *Variation 0*. Different values for β_P and β_N are tested.

Values for β_P and β_N	Positive score (PS)	Negative score (NS)
$\beta_P = 0$, $\beta_N = 0$ (baseline)	0.8861	0.8956
$\beta_P = 0.25$, $\beta_N = 0.25$	0.7878	0.7424
$\beta_P = 0.5$, $\beta_N = 0.5$	0.7916	0.5158
$\beta_P = 1$, $\beta_N = 1$	0.7996	0.5879
$\beta_P = 10$, $\beta_N = 10$	0.8495	**0.4733**
$\beta_P = 50$, $\beta_N = 50$	**0.9479**	0.6009
$\beta_P = 100$, $\beta_N = 100$	0.9325	0.6440

4.4 Evaluation Based on Visualization

We focus on the comparison of embeddings generated using *word2vec* (baseline), *Variation 0* of our enhanced embeddings, and *Variation 4*. In order to plot over the space the vectors of the embeddings generated (see Fig. 1), we performed dimensionality reduction, from the original 200 dimensions to 2, through Principal Component Analysis (PCA) over the vectors of the terms in Table 2 for the embeddings generated with these three representations. We focused over the embeddings representing the positive and negative predictive terms. For the resulting embeddings of our method (*Variation 0*), we selected β_P=50 and β_N=50 as parameter values.

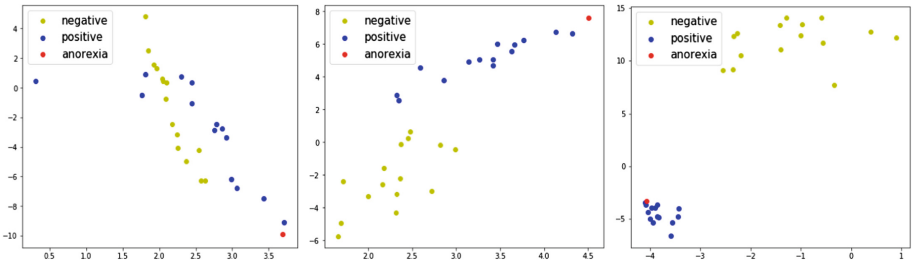

Fig. 1. Predictive terms sample represented on two dimensions after PCA was applied on their embeddings as dimensionality reduction method. From left to right each plot shows the vectorial representation of the predictive terms according to the embeddings obtained through (1) *Word2vec* (baseline), (2) *Variation 0*, and (3) *Variation 4*.

The positive predictive terms representations are closer after applying our method (*Variation 0*), and the negative predictive terms are displayed farther, in comparison to the baseline. The last plot displays the terms for the embeddings generated through Variation 4. For this case, given the input format for the retrofitting method, *anorexia* was linked with all the remaining predictive terms of the anorexia class (901), and likewise, each of these predictive terms was linked to the term *anorexia*. Notice that the retrofitting approach converges to changes in Euclidean distance of adjacent vertices, whereas the closeness between terms for our approach is given by the cosine distance.

4.5 Evaluation Based on the Predictive Task

In order to test our generated embeddings for the classification task dedicated to AN screening, we conduct a series of experiments to compare our method with related approaches. We define 5 baselines for our task: the first one is a BoW model based on word level unigrams and bigrams (*Baseline 1*), this model is kept mainly as a reference since our main focus is to evaluate our approach compared to other word embedding based models. We create a second model using *GloVe*'s pre-learned embeddings (*Baseline 2*), and a third model that uses word embeddings learned on the training set with the *Word2vec* approach (*Baseline 3*). We evaluate a fourth approach (*Baseline 4*) given by the enhancement of the *Baseline 3* embeddings, with Faruqui's *et al.* [7] retrofitting method. *Baseline 5* uses the same retrofitting method over GloVe's pre-learned embeddings, as we expected that a domain adaptation of the embeddings learned on a external source could be achieved this way.

Predictive Models Generation. To create our predictive models, again, each user is an instance represented by their writings (see Sect. 4.2). For *Baseline 1* we did a $tf \cdot idf$ vectorization of the users' documents, by using the *TfIdfVectorizer* provided by the *Scikit-learn* Python library, with a stop-words list and

the removal of the n-grams that appeared in less than 5 documents. The representation of each user through embeddings was given by the aggregation of the vector representations of the words in the concatenated texts of the users, normalized by the size (words count) of the document. Then, an L_2 normalization was applied to all the instances.

Given the reduced amount of anorexia cases on the training set, we used SMOTE [4] as an over-sampling method to deal with the unbalanced classes. The Scikit learn's Python library implementations for Logistic regression (LR), Random Forest (RF), Multilayer Perceptron (MLP), and Support Vector Machines (SVM) were tested as classifiers over the training set with a 5-fold cross validation approach. A grid search over each method to find the best parameters for the models was done.

Results. The results of the baselines are compared to models with our variations. For *Variation 4* and baselines 4 and 5 we use the 901 predictive terms of Sect. 4.4. To define the parameters of *Variation 3*, we test different configurations, as on Sect. 4.3, and chose the ones with the best results according to PS.

Precision (P), Recall (R), F1-Score ($F1$) and Accuracy (A) are used as evaluation measures. The scores for P, R and F1 reported over the test set on Table 4 correspond to the Anorexia (positive) class, as this is the most relevant one, whereas A corresponds to the accuracy computed on both classes. Seeing that there are 6 times more control cases than AN and that false negative (FN) cases are a bigger concern compared to false positives, we prioritize R and F1 over P and A. This is done because as with most medical screening tasks, classifying a user at risk as a control case (FN) is worst than the opposite (FP), in particular on a classifier that is intended to be a first filter to detect users at risk and eventually alert clinicians, who are the ones that do an specialized screening of the user profile. Table 4 shows the results for the best classifiers. The best scores are highlighted for each measure.

Comparing the baselines, we can notice that the embeddings based approaches provide an improvement on R compared to the BoW model, however this is given with a significant loss on P.

Regarding the embeddings based models, our variations outperform the results obtained by the baselines. The model with the embeddings generated with our method (*Variation 0*) provides significantly better results compared to the *Word2vec* model (*Baseline 3*), and even the model with pre-learned embeddings (*Baseline 2*), with a wider vocabulary.

The combination of pre-learned embeddings and embeddings learned on our training set, provide the best results in terms of F1 and R. They also provide a good accuracy considering that most of the test cases are controls. We can also observe that using the weights of pre-learned embeddings (*Variation 3*) to start our learning process over our corpus improves significantly the R score in comparison to *Word2vec*'s generated embeddings (*Baseline 3*).

The worst results for our variations are given by *Variation 1* that obtains equivalent results to *Baseline 2*. The best model in terms of F1 corresponds to

Variation 2. Also, better results are obtained for P when the embeddings are enhanced by the retrofitting approach (*Variation 4*).

Table 4. Baselines and enhanced embeddings evaluated in terms of Precision (P), Recall (R), F1-Score ($F1$) and Accuracy (A).

Model	Description	P	R	$F1$	A	Classifier
Baseline 1	BoW Model	**90.00%**	65.85%	76.06%	**94.69%**	MLP
Baseline 2	GloVe's pre-learned embeddings	69.57%	78.05%	73.56%	92.81%	MLP
Baseline 3	Word2vec embeddings	70.73%	70.73%	70.73%	92.50%	SVM
Baseline 4	Word2vec retrofitted embeddings	71.79%	68.29%	70.00%	92.50%	SVM
Baseline 5	GloVe's pre-learned embeddings retrofitted	67.35%	**80.49%**	73.33%	92.50%	MLP
Variation 0	Predictive pairs embeddings ($\beta_P = 50$ $\beta_N = 50$)	77.50%	75.61%	76.54%	94.03%	MLP
Variation 1	Predictive pairs embeddings + GloVe embeddings	69.57%	78.05%	73.56%	92.81%	MLP
Variation 2	Predictive pairs embeddings ($\beta_P = 50$ $\beta_N = 50$) + GloVe embeddings	75.00%	**80.49%**	**77.65%**	94.06%	MLP
Variation 3	Predictive pairs embeddings + GloVe embeddings starting weights ($\beta_P = 0.25$ $\beta_N = 50$)	72.73%	78.05%	75.29%	93.44%	MLP
Variation 4	Predictive pairs ($\beta_P = 50$ $\beta_N = 50$) retrofitted embeddings	82.86%	70.73%	76.32%	94.37%	SVM

5 Conclusions and Future Work

We presented an approach for enhancing word embeddings towards a classification task on the detection of AN. Our method extends *Word2vec* considering positive and negative costs for the objective function of a target term. The costs are added by defining predictive terms for each of the target classes. The combination of the generated embeddings with pre-learned embeddings is also evaluated. Our results show that the usage of our enhanced embeddings outperforms the results obtained by pre-learned embeddings and embeddings learned through *Word2vec* regardless of the small size of the corpus. These results are promising as they might lead to new research paths to explore.

Future work involves the evaluation of the method on similar tasks, which can be formalized as document categorization problems, addressing small corpora. Also, ablation studies will be performed to assess the impact of each component into the results obtained.

References

1. Arseniev, A., Lee, H., McCormick, T., Moreno, M.: Proana: pro-eating disorder socialization on twitter. J. Adolesc. Health **58**, 659–664 (2016)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017)
3. Çano, E., Morisio, M.: Word embeddings for sentiment analysis: a comprehensive empirical survey. CoRR abs/1902.00753 (2019)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
5. Coppersmith, G., Leary, R., Crutchley, P., Fine, A.: Natural language processing of social media as screening for suicide risk. Biomed. Inform. Insights **10** (2018)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018)
7. Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, E., Smith, N.A.: Retrofitting word vectors to semantic lexicons. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 1606–1615. Association for Computational Linguistics (2015)
8. Guntuku, S.C., Yaden, D.B., Kern, M.L., Ungar, L.H., Eichstaedt, J.C.: Detecting depression and mental illness on social media: an integrative review. Curr. Opin. Behav. Sci. **18**, 43–49 (2017)
9. Kuang, S., Davison, B.D.: Learning word embeddings with chi-square weights for healthcare tweet classification. Appl. Sci. **7**(8), 846 (2017)
10. Losada, D.E., Crestani, F., Parapar, J.: Overview of eRisk: early risk prediction on the internet. In: Bellot, P., et al. (eds.) CLEF 2018. LNCS, vol. 11018, pp. 343–361. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98932-7_30
11. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)
12. Mowafy, M., Rezk, A., El-Bakry, H.: An efficient classification model for unstructured text document. Am. J. Comput. Sci. Inf. Technol. **06**, 16 (2018)
13. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics (2014)
14. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of NAACL (2018)
15. Tissier, J., Gravier, C., Habrard, A.: Dict2vec : learning word embeddings using lexical dictionaries. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 254–263. Association for Computational Linguistics, Copenhagen, September 2017
16. Yin, W., Schütze, H.: Learning word meta-embeddings. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1351–1360. Association for Computational Linguistics, Berlin, August 2016
17. Zhang, Y., Li, H.J., Wang, J., Cohen, T., Roberts, K., Xu, H.: Adapting word embeddings from multiple domains to symptom recognition from psychiatric notes. In: AMIA Joint Summits on Translational Science Proceedings. AMIA Joint Summits on Translational Science (2018)