Two Simple and Domain-independent Approaches for Early Detection of Anorexia

Sergio Burdisso, Leticia Cagnina, Marcelo Errecalde, and Manuel Montes-y-Gómez

Abstract In this chapter, we describe the participation of our research team in the eRisk addressing the two editions of the early anorexia detection task. We used two domain-independent approaches to address this task. The first approach is based on a temporal-aware document representation, whereas the second one consists of a simple, interpretable, and novel text classification model specially designed for addressing early risk detection scenarios. Regarding the obtained results, in the first edition, we achieved the best $ERDE_5$ value among all participant models using the first approach, whereas with the second one, the best precision (0.91). Besides, using the latter approach, in the second edition, we were able to achieve the best values for both $ERDE_5$ and $ERDE_{50}$, and also promising results in terms of the ranking-based metrics, obtaining the best values, consistently, across all four rankings.

S. Burdisso (✉) · L. Cagnina · M. Errecalde
Universidad Nacional de San Luis (UNSL), San Luis, Argentina
e-mail: sburdisso@unsl.edu.ar; sergio.burdisso@idiap.ch

L. Cagnina
e-mail: lcagnina@unsl.edu.ar

M. Errecalde
e-mail: merreca@unsl.edu.ar

S. Burdisso
Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland

L. Cagnina
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Buenos Aires, Argentina

M. Montes-y-Gómez
Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, Mexico
e-mail: mmontesg@inaoep.mx

F. Crestani et al. (eds.), *Early Detection of Mental Health Disorders by Social Media Monitoring*, Studies in Computational Intelligence 1018,
https://doi.org/10.1007/978-3-031-04431-1_7

1 Introduction

The term "mental health problem" comprises a broad spectrum of disorders that can be grouped into different categories, being "eating disorders" one of them [22]. Anorexia and bulimia are the most common eating disorders in childhood and adolescence years [7]. Both are characterized by an overvaluation of the weight and body shape such that even when people are underweight they try to avoid weight gain [8]. Anorexia is a severe psychiatric disorder with the highest mortality of all mental disorders [9] but can be worse in people facing a particular situation such as a pandemic. A recent survey showed that due to the COVID-19 pandemic, 62% of people in the U.S. with anorexia experienced a worsening of symptoms [20].

It is becoming increasingly common for people suffering from this kind of mental disease to share their own experience on social media, for instance, to seek support and containment. On the other hand, the increased use of social media and the advancement of computational technologies allow the extraction of valuable information to early prevent risky situations. In particular, early risk detection (ERD) on social media has become an important research area that is gaining increasing popularity [12–15] due to the potential impact it could have in people's lives—since early detection systems could help at-risk people to get the care and social support they need on time, before it is too late. This fostered the creation of the eRisk lab, hosted annually at the CLEF congress since 2017 [13], in which research teams from all over the globe can participate by creating different models to address specific ERD tasks.

Our team, UNSL, has participated in four of the five editions of the eRisk lab held to date. Throughout the different editions, we have participated using mainly two simple approaches, one based on a temporal document representation, called FTVT [6], and the other, on a novel text classifier that was specially designed for ERD scenarios, called SS3 [2]. Using these two approaches our team has obtained consistently competitive results in the different tasks proposed at each eRisk edition. For instance, among all participating teams, we achieve the best ERDE_{50} value in the eRisk 2017's early depression detection task [5], the best ERDE_5 and precision in both early depression and anorexia detection tasks of the eRisk 2018 [6], the best ERDE_5 and ERDE_{50}, and the best overall ranking-based performance in both early anorexia and self-harm detection tasks of the eRisk 2019, and the best $F_{latency}$, F_1 and precision in the eRisk 2019's early self-harm detection task [3], and finally, the second-best ERDE_5 and $F_{latency}$ in the eRisk 2020's early self-harm detection task. Besides, using the SS3 classifier, we later obtained and published the best ERDE_5 and ERDE_{50} reported values for the eRisk 2017 early depression detection task [2, 4] and the early depression and anorexia detection tasks of the eRisk 2018 [4].

This chapter describes how we addressed the early anorexia detection task. It is organized as follows: Sect. 2 introduces our two main approaches mentioned above, respectively, the FTVT representation and the SS3 classifier. Then, Sect. 3 describes in detail our participation in the two editions of this task, where details are given about how our models were implemented, trained, fine-tuned, and finally evaluated,

as well as presenting and analyzing the obtained results. Finally, Sect. 4 summarizes the main conclusions and suggests possible future work.

2 Approaches

As with the other eRisk tasks, we used two different approaches to address the early anorexia detection task, one based on a document representation called FTVT, and the other, on a text classifier called SS3. Both FTVT and SS3, are introduced and described, respectively, in the following two sections.

2.1 Flexible Temporal Variation of Terms (FTVT)

As we described in the paper published on CLEF's eRisk 2018 track [6], the Flexible Temporal Variation of Terms (FTVT) is an improvement of the *temporal variation of terms* (TVT) document representation that we developed for the eRisk 2017 depression detection task [5]. The TVT representation is based on the *concise semantic analysis* (CSA) technique proposed by Li et al. [10]. CSA is a semantic analysis technique that interprets words in a space $\mathcal{C}$ of concepts that are close (or equal) to the category labels. For instance, if documents in the dataset are labeled with k different category labels, documents will be represented in a k-dimensional space. That space size is usually much smaller than standard BoW[23] representations which directly depend on the vocabulary size (usually much larger than the number of categories). CSA has been used in common text categorization tasks [10] and has been adapted to work in author profiling tasks under the name of *Second-Order Attributes* (SOA) [11].

More precisely, the TVT representation assumes we are dealing with chunk-based early risk detection scenarios as the ones originally proposed by Losada et al. [12]—and later used in the first two editions of the eRisk lab [13, 14]. That is, it assumes a corpus of documents written by p different individuals ($\{I_1, \ldots, I_p\}$). For each individual I_i ($i \in \{1, \ldots, p\}$), the complete chronological sequence of his/her d_i documents ($D_{I_i,1}, D_{I_i,2}, \ldots, D_{I_i,d_i}$) is split into n different "chunks", $C_{I_i,1}, C_{I_i,2}, \ldots, C_{I_i,n}$—therefore, $C_{I_i,1}$ contains the first $100/h\%$ of the documents of the individual I_i, chunk $C_{I_i,2}$ contains the second $100/h\%$, and so on. Then, models have to process those p individuals as p different sequences of n chunks and make a binary decision (as early as possible) on whether or not each individual might be a positive case of risk. Note that, as shown in Fig. 1a, using a standard CSA representation under this scenario will represent each individual I_i as a vector of only two concepts—since the decision is binary, there are only two category labels, *positive* (risk case) and *negative* (non-risk case). Instead, the TVT representation uses the additional temporal information available under this scenario to obtain an improved concept space. The underlying idea is that the variations of the terms (e.g. words)

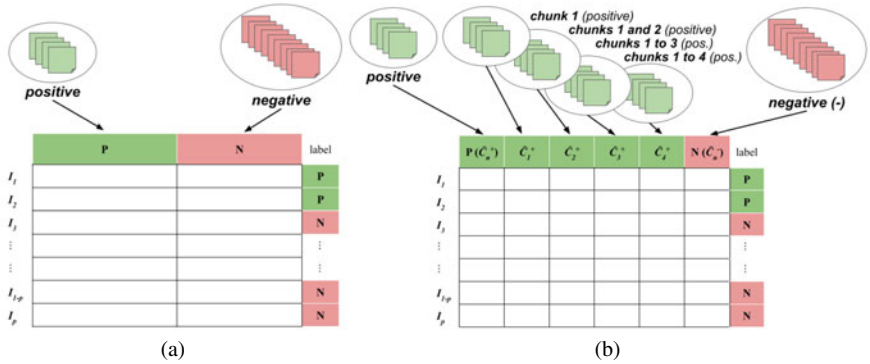

Fig. 1 Comparison of representations for early risk detection tasks: **a** Concise Semantic Analysis (CSA); **b** Temporal Variation of Terms (TVT) with $k = 4$

used in different sequential (temporal) stages can provide relevant information to the classification model. As a consequence, this representation enriches the documents of the minority *at-risk* class by including information of documents read in the first k chunks, as new CSA concepts in the original concept space. More precisely, in symbols, let $\hat{C}_j$ be the subset of the dataset containing only the first j chunks of each individual, $\hat{C}_j^+$ and $\hat{C}_j^-$ a partition of $\hat{C}_j$ in which $\hat{C}_j^+$ only contains the positive at-risk individuals and $\hat{C}_j^-$ the negative ones, then TVT uses the subsets $\hat{C}_1^+, \hat{C}_2^+, \ldots, \hat{C}_k^+$ as well as $\hat{C}_n^+$ (i.e. the complete positive dataset) as different concepts for the positive class and $\hat{C}_n^-$ (i.e. the complete negative dataset) for the negative class—as illustrated in Fig. 1b. Note that TVT not only enriches the original time-unaware representation with temporal information but also allows to address the unbalance of the minority class by augmenting it with this extra information. Preliminary results showed the potential of this representation, in comparison to CSA and BoW representations, to deal with ERD problems under this type of scenario [5]. Thus, when working with chunk-based early classification scenarios, as in the first two editions of the eRisk [13, 14], TVT naturally copes with both the sequential characteristics of ERD problems and also helps to deal with the class imbalance problem.

Finally, when we originally proposed this representation [5], the base CSA representation of the minority class was augmented by adding 4 extra dimensions. However, we then observed and empirically verified that, as expected, by varying this number, different performance was achieved depending on the specific o used for the ERDE_o measure to evaluate the results. Therefore, FTVT, the variant of TVT used in the present work, is simply a "flexible version" of the original TVT in which the user can specify the number k of chunks to be used to create the representation of the minority *at-risk* class. This number allows maximizing the model's performance according to the urgency level required for the specific ERD task (which is determined by the ERDE's threshold o). However, beyond this small distinction, TVT and FTVT are conceptually the same approaches.

2.2 SS3 Text Classifier

The SS3 text classifier is a novel classification model that we originally proposed in Burdisso et al. [2]. This model was specially designed and created with the goal of addressing early risk detection problems as naturally and integrally as possible. To achieve this, first, we noticed that these types of problems are especially challenging for conventional models because tackling them, as a whole, involves at least three key aspects: *sequence classification*, *early classification*, and also, given the sensitive nature of the problem, *model transparency and interpretability*. Then, we identified three key capabilities our model should have to tackle each of those three key aspects, respectively: *being able to work incrementally*, *provide support for early decision-making mechanisms*, and *having the ability to visually explain the reasons behind their predictions*. Finally, the SS3 text classifier was the result of our attempt to create a simple model that was able to integrate those three capabilities as a whole, naturally, by design. The following two sections briefly describe the model's training and the classification process, respectively.

2.2.1 Training

The SS3's training process is trivial since it only consists of building, for each given category, a dictionary to store word frequencies using all training documents of the given category. This simple training process allows the model to support online learning since when new training documents are available, instead of training again from scratch, the model only needs to update the dictionaries using the content of the new documents, making the training incremental.

Using the word frequencies stored in the dictionaries, the model computes a value for each word using a function, called gv,[1] to value words in relation to categories. This gv function can be computed "on demand" during classification or computed (and cached) as part of the training process. It takes a word w and a category c and returns a number, in the interval [0, 1], representing the degree of confidence with which w is believed to "be important" to c. Besides, since interpretability needs to be defined in a domain-specific way [18], we first defined what constituted interpretability by considering how people could explain to each other the reasoning processes behind a typical text classification tasks,[2] and then this gv function was designed to value words by trying to mimic that behavior –i.e. having gv to value words "the way people would intuitively do it". For instance, suppose categories are

[1] The name gv stands for "global value".

[2] As it turns out, for text classification, people would normally direct our attention only to certain "keywords" (filtering out all the rest) and explain why these words were important in their reasoning process.

$C = \{food, music, health, sports\}$, then, after training, SS3 would learn to assign values like:

$gv(\text{'sushi'}, food) = 0.85;$	$gv(\text{'the'}, food) = 0;$
$gv(\text{'sushi'}, music) = 0.09;$	$gv(\text{'the'}, music) = 0;$
$gv(\text{'sushi'}, health) = 0.65;$	$gv(\text{'the'}, health) = 0;$
$gv(\text{'sushi'}, sports) = 0.02;$	$gv(\text{'the'}, sports) = 0;$

To achieve this, the actual computation of gv is carried out by the product of three functions, lv, sg, and sn, respectively. In symbols:

$$gv(w, c) = lv_\sigma(w, c) \cdot sg_\lambda(w, c) \cdot sn_\rho(w, c) \tag{1}$$

The intuition captured by these functions can be briefly summarized as follows:

- $lv_\sigma(w, c)$ values a word based on the local frequency of w within c. As part of this process, the word distribution curve can be smoothed by a factor controlled by the hyperparameter σ.
- $sg_\lambda(w, c)$ captures the importance of w in relation to c. It is a sigmoid function that returns a value close to 1 when $lv(w, c)$ is "significantly greater" than $lv(w, c_i)$, for most of the other categories c_i; and a value close to 0 when, for all c_i, $lv(w, c_i)$ values are close to each other. The λ hyperparameter controls the "significantly greater" part, i.e. it allows to control how far $lv(w, c)$ must deviate from the median of all $lv(w, c_i)$ to be considered significant.
- $sn_\rho(w, c)$ decreases the final value in relation to the number of categories w is significant to. That is, the more categories c_i for which $sg_\lambda(w, c_i) \approx 1$, the smaller the $sn_\rho(w, c)$ value. The ρ hyperparameter controls how severe this sanction is.

As it is described in more detail by Burdisso et al. [2], the actual equation for each function was deduced by trying to capture a particular aspect/intuition involved in having gv assign a final value that, as mentioned above, matched our interpretability criteria. These equations are, respectively, the following[3]:

$$lv_\sigma(w, c) = \left(\frac{tf_{w,c}}{max\{tf_c\}}\right)^\sigma \tag{2}$$

where $tf_{w,c}$ denotes the frequency of w in c, $max\{tf_c\}$ the maximum frequency seen in c;

$$sg_\lambda(w, c) = \frac{1}{2} tanh\left(4\frac{(lv(w, c) - \widetilde{LV}_w)}{\lambda \cdot MAD_w} - 2\right) + \frac{1}{2} \tag{3}$$

where $LV_w = \{lv(w, c_i) | c_i \in C\}$, $\widetilde{LV}_w$ denotes the median of LV, $MAD_w = median(|lv(w, c_i) - \widetilde{LV}_w|)$ i.e. the *Median Absolute Deviation* of LV_w;

[3] Readers interested in knowing how these equations were determined are invited to read Sect. 3 of the original paper [2].

$$san_{\rho}(w,c) = \left(\frac{|C| - \left(\hat{C}_{wc} + 1\right)}{(|C| - 1)\left(\hat{C}_{wc} + 1\right)} \right)^{\rho} \tag{4}$$

where $|C|$ denotes the total number of categories and $\hat{C}_{wc}$ the accumulated "significance" of w across all the categories except c, in symbols:

$$\hat{C}_{wc} = \sum_{c_i \in C - \{c\}} sig_{\lambda}(w, c_i)$$

2.2.2 Classification

Before describing the overall classification process, we will introduce a vectorial version of the gv function since the classification process makes use of it. Namely, we will simply define $\overrightarrow{gv}$ as $\overrightarrow{gv}(w) = \langle gv(w, c_0), gv(w, c_1), \ldots, gv(w, c_k)\rangle$ where $c_i \in C$ and C is the set of all given categories. That is, $\overrightarrow{gv}$ takes a word and outputs a vector in which each component is the word's gv for each category c_i. Thus, for instance, given the previous example, we would have:

$\overrightarrow{gv}(\text{'}sushi\text{'}) = \langle 0.85, 0.09, 0.65, 0.02\rangle$;

$\overrightarrow{gv}(\text{'}the\text{'}) = \langle 0, 0, 0, 0\rangle$;

These vectors are called "confidence vectors" and, therefore, in the above example $\langle 0.85, 0.09, 0.65, 0.02\rangle$ is the *confidence vector* of the word "sushi" and $\langle 0, 0, 0, 0\rangle$ the *confidence vector* of "the".

As illustrated with an example in Fig. 2, the classification process can be thought of as a 2-phase process. In the first phase, the input is split into multiple blocks (e.g., paragraphs), then each block is in turn repeatedly divided into smaller units (e.g., sentences, words). Thus, the previously "flat" document is transformed into a hierarchy of blocks. In the second phase, the $\overrightarrow{gv}$ function is applied to each word to obtain the "level 0" *confidence vectors*, which then are reduced to "level 1" *confidence vectors* by means of a level 0 *summary operator*, $\oplus_0$.[4] This reduction process is recursively propagated up to higher-level blocks, using higher-level *summary operators*, $\oplus_j$, until a single *confidence vector*, $\overrightarrow{u}$, is generated for the whole input. Finally, the actual classification is performed based on the values of this single *confidence vector*, $\overrightarrow{u}$, using some policy—for example, selecting the category with the highest *confidence value*.

This process allows the model to possess the three desired capabilities: (a) keeping track of how the final vector $\overrightarrow{u}$ changes over time allows the model to *provide support for early decision-making mechanisms*, i.e. it allows deriving simple and clear rules to decide *when* the system should stop and make an early classification; (b) the model can visually explain the reasons behind the classification by painting, hierarchically, the different parts of the input in proportion to the values of the different *confidence*

[4] Any function $f : 2^{\mathbb{R}^n} \mapsto \mathbb{R}^n$ could be used as a *summary operator*. In this example, vector addition was used for $\oplus_1$ but not for $\oplus_0$ to highlight this fact.

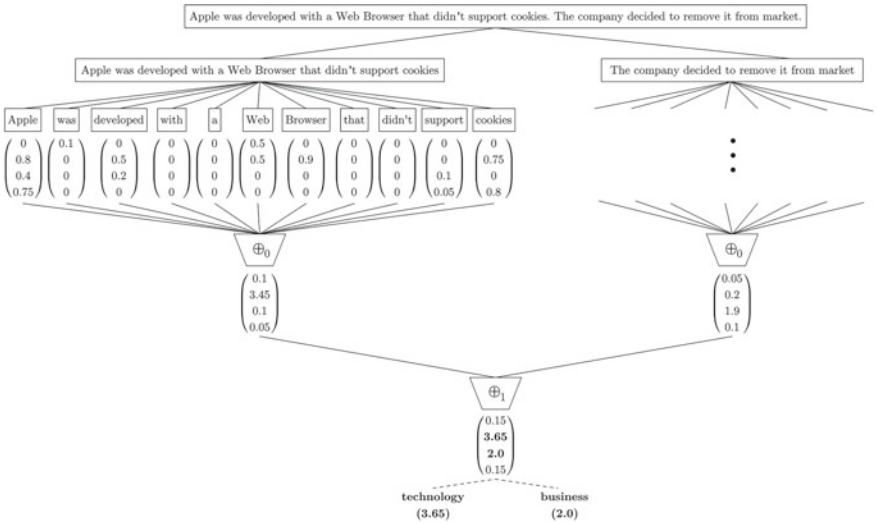

Fig. 2 Classification process for a hypothetical example document "Apple was developed with a Web Browser that didn't support cookies. The company decided to remove it from market". In the first stage, this document is split into two sentences (for instance, by using the dot as a delimiter) and then each sentence is split into single words. In the second stage, *global values* are computed for every word to generate the first set of *confidence vectors*. Then, all these word vectors are reduced by the $\oplus_0$ operator to sentence vectors, $\langle 0.1, 3.45, 0.1, 0.05\rangle$ and $\langle 0.05, 0.2, 1.9, 0.1\rangle$ for the first and second sentence respectively. After that, these two sentence vectors are reduced by another operator ($\oplus_1$, which in this case is the addition operator) to a single *confidence vector* for the entire document, $\langle 0.15, 3.65, 2.0, 0.15\rangle$. Finally, a policy is applied to this vector to make the classification—which in this example was to select *technology*, the category with the highest value, and also *business* because its value was "close enough" to *technology*'s

vectors in the hierarchy[5]; (c) finally, this process also allows the model to work incrementally, so long as the *summary operator* for the highest level can be computed incrementally. For instance, suppose that later, a new sentence is appended to the example shown in Fig. 2. Since $\oplus_1$ is the vector addition, instead of processing the whole document again, the already computed vector, $\langle 0.15, 3.65, 2.0, 0.15\rangle$, is simply updated by adding the new sentence *confidence vector* to it.

[5] A live demo is provided at http://tworld.io/ss3where the interested readers can try out the model online. Along with the classification result, the demo provides an interactive visual explanation as the one suggested here. We believe explanations like these are vital when models' predictions could affect people's lives since it allows human experts to inspect the reasons behind the classifications and validate them [Las accessed date: April 2021].

3 Participation and Results

In this section, we show the details of our participation addressing the eRisk's early detection of anorexia tasks. We first describe the experimental settings, then we present the results obtained after the evaluation stage. Our team participated in the two editions of this task, eRisk 2018 [6] and 2019 [3]. Since the evaluation was performed differently in the two editions,[6] this section is organized in two main subsections, one for each one.

3.1 Early Detection of Anorexia—2018 Edition

The 2018 edition had the participation of 9 different research teams and a total of 34 models were submitted. As described in more detail in our eRisk paper [6], we participated with five models, three of them made use of the FTVT representation (referred as to UNSLA, UNSLB, and UNSLC) and the other two (UNSLD and UNSLE), used the SS3 classifier.

The three models using the FTVT representation used the same early classification policy which was based on the probability p assigned by the classifiers. More precisely, as soon as the probability exceeded some given threshold θ (i.e. $p \geq \theta$), the subject was classified as positive (at risk). Since we also participated in the early depression detection task of this eRisk edition, to obtain more confident statistics, we used only the depression training set for setting the parameters of our models because it was the largest. Thus, in order to find the best values for this threshold θ, as well as the other parameters, we performed a *stratified 5-fold cross-validation* on the depression training set and then we carried out two grid searches, one minimizing the ERDE_5 measure, and the other the ERDE_{50}. In addition to optimizing the probability threshold θ, the grid search also evaluated different values for the FTVT's n parameter (number of chunks used to enrich the representation of the positive class), as well as different classification models. Namely, we considered values ranging from 0 to 5 for the n parameter, the values 0.6, 0.7, 0.8 and 0.9 for the threshold θ, and different classification models such as Logistic Regression (LR), Support Vector Machine (SVM), and Naive Bayes (NB)—these models were coded in Python 2.7 using the implementation provided by the *Scikit-learn* package with the default parameters. Finally, from the results obtained after the grid searches we selected the following 3 models for participating:

- *UNSLA*: an SVM classifier using the FTVT representation with $n = 0$ and a probability threshold $\theta = 0.8$. This model was selected because it obtained the best ERDE_5 of 13.58 compared to the second-best model which achieved 13.68. Note

[6] For instance, in the first edition [13] the release of the evaluation data was chunk-by-chunk whereas, in the second edition [14], user content was released post by post. Additionally, a new set of evaluation metrics was used.

that, from Sect. 2.1, this $n = 0$ means that no enrichment of the positive class is performed in the FTVT representation, and thus, the actual representation is identical to the standard CSA representation.
- *UNSLB*: a Logistic Regression classifier using the FTVT representation with $n = 2$ and a probability threshold $\theta = 0.6$. This model was selected because it obtained the best balance between ERDE_5 and ERDE_{50}.
- *UNSLC*: an SVM classifier using the FTVT representation with $n = 4$ and a probability threshold $\theta = 0.7$. This model was selected because it obtained the best ERDE_{50}.

As mentioned at the beginning of this section, the remaining 2 models, UNSLD and UNSLE, did not use the FTVT representation and instead used the SS3 classifier. In particular, these two models used a version of the SS3 classifier in which the vector addition was used as the *summary operator*, $\oplus_j$, for all the hierarchy levels and, as a consequence, the classification process was simplified to the following word summation:

$$\overrightarrow{u} = \sum_{w \in S} \overrightarrow{gv}(w) \tag{5}$$

where S is the subject's writing history. Note that, in this particular task, $\overrightarrow{u}$ is a vector with two components, one for the positive *at-risk* class (anorexic) and the other for the negative one. The early classification policy used to classify a subject as at-risk was performed by analyzing how $\overrightarrow{u}$ changed over time (i.e. over "chunks"). More precisely, subjects were classified as positive when the accumulated positive confidence value exceeded the negative one, as shown in Fig. 3. Regarding the hyperparameter values, they were manually selected using a subset of the training set as a validation set, from which we obtained the following 2 model configurations for participating:

- *UNSLD*: SS3 classifier with $\sigma = 0.5$, $\rho = 1$ and $\lambda = 3$.
- *UNSLE*: SS3 classifier with the same hyperparameter configuration as UNSLD but with $\lambda = 7$.

Since a word's value has to be 7 ($\lambda = 7$) times greater than the median to be considered by UNSLE, we expected this model to yield a higher precision than UNSLD with the risk of having a lower recall. Similarly, by having $\lambda = 3$, UNSLD was meant to consider a wider range of words as being "important", thus, we expected this model to have a higher recall compared to UNSLE.

Finally, we trained our 5 models using the entire training set available for this task and proceeded to the evaluation stage in which they were evaluated along with all the other 29 participant models. The results obtained by our 5 models, after the evaluation stage, are shown in Table 1. As we can see, UNSLB obtained the best ERDE_5 (11.40%) and UNSLD the best precision (0.91) value among all participant models. It is worth mentioning, again, that the parameters for the models using the FTVT representation were not specially selected for the anorexia task but for the depression task and yet, they obtained good results, especially UNSLB and UNSLC.

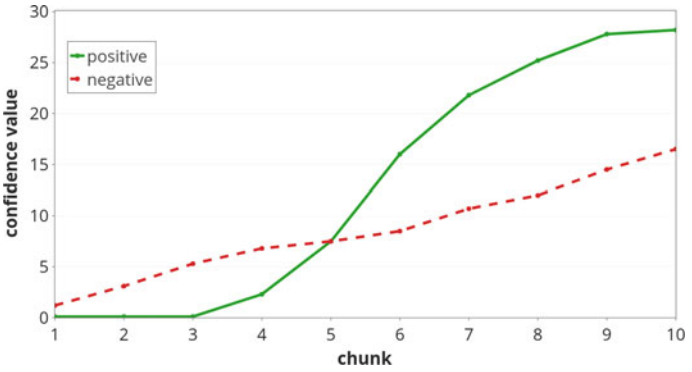

Fig. 3 Subject 9579's confidence vector $\overrightarrow{u}$ variation over time (chunks). This plot shows how the positive and negative values stored in $\overrightarrow{u}$ changed as more chunks were processed for this subject. For instance, only after processing the 3rd chunk, the positive confidence started to grow, probably because this subject was talking about topics not related to anorexia up until that point. Finally, after the 5th chunk, the positive confidence outweighed the negative one and the subject is classified as "anorexic"

Table 1 Results obtained by our five models on the eRisk 2018 early anorexia detection task. For comparison, the models that obtained the best results for the other 7 participating teams are also included

Model	$ERDE_5$ (%)	$ERDE_{50}$ (%)	R	P	F_1
UNSLA	12.48	12.00	0.10	0.57	0.17
UNSLB	**11.40**	7.82	0.51	0.75	0.61
UNSLC	11.61	7.82	0.51	0.75	0.61
UNSLD	12.93	9.85	0.71	**0.91**	0.79
UNSLE	12.93	10.13	0.63	0.90	0.74
FHDO-BCSGD	12.15	**5.96**	**0.88**	0.75	0.81
FHDO-BCSGE	11.98	6.61	0.83	0.87	**0.85**
PEIMEXB	12.41	7.79	0.73	0.57	0.64
RKMVERIA	12.17	8.63	0.56	0.82	0.67
LIIRB	13.05	10.33	0.73	0.79	0.76
LIIRA	12.78	10.47	0.63	0.81	0.71
TBSA	13.65	11.14	0.76	0.60	0.67
UPFA	13.18	11.34	0.71	0.74	0.72
UPFD	12.93	12.30	0.46	0.86	0.60
TUA1C	13.53	12.57	0.32	0.42	0.36
LIRMMB	14.45	12.62	0.71	0.41	0.52
LIRMMA	13.65	13.04	0.56	0.52	0.54

UNSLA did not obtain competitive results and its performance was the lowest among all participating models. This suggests that, in fact, the "temporal enrichment" performed by the FTVT representation does have a considerable positive impact on the classifier performance.[7]

Regarding the two models using the SS3 classifier, as we expected, UNSLE obtained a recall lower than UNSLD's since it considers fewer words as being important—only words whose values are 7 times greater than the median. However, against our expectation, UNSLE's precision was also lower than UNSLD's—we were expecting a higher precision given that words were more strictly selected. This suggests that $\lambda = 3$ represents a good balance between recall and precision, and that increasing its value further, i.e. being more strict at selecting words, only yields a lower recall. Among our 5 models, both SS3 models achieved a better performance than the 3 FTVT models in terms of the standard (timeless) measures but were considerably less efficient in terms of the ERDE measures. However, we later discovered that we could have also obtained a better performance in terms of the ERDE measures with the SS3 classifier if we had used the same hyperparameter configuration as the one used in the chunk-based experimental scenario for the early depression detection task in the SS3's original paper [2]. More precisely, as we published later [4], an SS3 classifier with $\lambda = \rho = 1$ and $\sigma = 0.45$ would have outperformed all three FTVT-based models by achieving a better performance in term of standard measures (0.71, 0.77 and 0.66 for F_1, precision and recall, respectively) as well as in terms of both ERDE measures—obtaining the best ERDE_5 value (11.31%) and the second-best ERDE_{50} value (6.26%) among all participating models.

Finally, it is interesting to mention that the FHDO-BCSGD model was a Convolutional Neural Network (CNN) with an architecture consisting of 100 filters with 2 feature maps, and a max-pooling layer followed by 4 fully connected layers. In addition, the input of this network consisted of 6 million different word embeddings (fastText) manually trained using a dataset of 1.7 billion Reddit comments specifically built for this task [21]. On the other hand, FHDO-BCSGE consisted of a late fusion ensemble of other FHDO-BCSG's models, namely, an ensemble of two CNN models (one of which is FHDO-BCSGD) with a model that, in turn, is an ensemble of 4 Logistic Regressions with different BoW representations augmented with hand-crafted domain-specific metadata [21]. The complexity of these models contrasts with the simplicity of ours which, besides using simpler classifiers, did not make use of any external resources besides the available training data. For instance, UNSLD consisted only of a simple SS3 classifier[8] and yet its performance in terms of standard measures (F_1, precision and recall) were comparable to the more complex FHDO-BCSG's models described above—i.e. despite its simplicity, this SS3 model not only achieved the best precision (0.91) among the participating models but also

[7] Note that, unlike UNSLB and UNSLC, UNSLA used FTVT with $n = 0$ and, therefore, the actual representation was identical to a standard CSA representation—with no temporal chunk-based enrichment of the positive class.

[8] Which simply consisted of a summation of word values learned from the available training data (Eq. 5).

obtained an F_1 value (0.79) that was only outperformed by the considerably more complex FHDO-BCSG's models (0.81 and 0.85).

3.2 Early Detection of Anorexia—2019 Edition

As described in more detail in the overview of the 2019 edition of the eRisk [15], the task was intended as a continuation of the previous edition, so even though a new test set was built, the training set was formed by joining the 2018 Edition's training and test sets. As indicated at the beginning of this section, the evaluation methodology was different from the previous one since users were no longer processed using the chunk-by-chunk approach. Instead, they were processed in a more realistic way, one post (writing) at a time. Also, performance evaluation was improved since, in addition to reporting the time taken for each team to complete the task, two different types of metrics were used. The first type was focused on early classification decisions, and the second on rankings of users by risk level (estimated by the models). More precisely, in addition to the ERDE measure, this edition also incorporated the $F_{latency}$ [19] measure as an extra decision-based metric. Regarding the ranking-based metrics, to create the rankings for evaluating each model, after processing each user post, models were asked to respond with a score representing the risk level estimated for each user up to that moment. Then, for each participating model, the ranking was created by ordering the users using the given score. Finally, the quality of the rankings was evaluated using two standard IR metrics, $P@k$, and $NDCG@k$. Specifically, models were evaluated in terms of $P@10$, $NDCG@10$, and $NDCG@100$ for 4 different rankings, the one obtained after processing, respectively, 1, 100, 500, and 1000 user posts.

The 2019 edition had the participation of 13 different research teams and a total of 50 models were submitted. As described in more detail in our eRisk paper [3], in this edition we also participated with five models. However, due to this edition replacing the chunk-based release of users data by a more realistic post-by-post one, unlike in the previous edition, none of our five models used the FTVT representation and, instead, all of them were implemented using the SS3 classifier. This decision was motivated by the promising results previously obtained with SS3 [2, 4] as well as by the fact that, unlike the FTVT representation that requires the input data being released using the chunk-by-chunk approach, SS3 was designed to naturally work over text streams and, as a consequence, it can naturally process data incrementally regardless of the specific approach being used to release it—e.g. chunk-by-chunk, post-by-post, word-by-word, and so on.

In this edition we again made use of the same simple version of the SS3 classifier as in the 2018 edition (Eq. 5) and the same simple early classification policy—i.e. subjects were classified as positive as soon as the accumulated positive confidence value exceeded the negative one. However, this time we did use the hyperparameter configuration with which, as mentioned in the previous section, we could have obtained the best ERDE values in the 2018 edition—i.e. we used $\lambda = \rho = 1$ and

$\sigma = 0.45$ for our five SS3 models. We decided to use this configuration for all five models because in previous experiments they showed to be very effective and robust in terms of the ERDE measures [2, 4].[9] Finally, since we used the same classifier with the same hyperparameter configuration for our five models, the main difference among them was given by the amount of data used for the training and whether the model took into account only words or also bigrams, as described below:

- *UNSL#0*: the model was trained only with a subset of the total training data available for this edition. Particularly, the training was performed using only the data equivalent to the training set of the 2018 edition. The idea behind this model was to evaluate the consistency and robustness of results obtained in that edition.
- *UNSL#1*: this model was the same as the previous one but allowing SS3 to also recognize word bigrams—i.e. SS3 learned to compute $gv(w_0, c)$ as well as $gv(w_0 w_1, c)$ for each w_0, w_1 seen during training. This variant was driven by good preliminary results obtained by SS3 using variable-length word n-grams with data from previous eRisk editions [4].
- *UNSL#2*: this model was, again, the same as UNSL#0 but this time, the training was performed using all available data for this edition—i.e. joining training set and test sets from the 2018 edition.
- *UNSL#3*: this model was the same as UNSL#2 but, as with UNSL#1, allowing SS3 to also recognize word bigrams.
- *UNSL#4*: this model was a slightly modified version of the UNSL#2 model in which, during the classification process, only words with a *global value* greater than 0.3 were taken into account—i.e. SS3 assigned $gv(w, c_i) = 0$ to all w and c_i such that $gv(w, c_i) \leq 0.3$.[10] With this variant, we tried to measure what impact words with a low *global value* have on classification performance. That is, with this model we intended to address questions such as: "How sensitive is the model to the noise that could be caused by a large number of words with low *global value*?" or "Could it be the case that because of words with low *global value* the early classification policy condition could be incorrectly triggered due to the noise accumulated by processing them?".

Besides, since in this edition, as described above, models were also asked to provide a score representing the risk level estimated for each user, we used the user *confidence vector*, $\overrightarrow{u}$ in Eq. 5, to calculate this score simply as the difference between the positive and negative *confidence value*, that is:

$$\text{score} = \overrightarrow{u}[positive] - \overrightarrow{u}[negative] \tag{6}$$

[9] This hyperparameter configuration was originally discovered with the eRisk 2017 early depression detection dataset by applying a grid search to minimize the ERDE_{50} metric with the training set using a 4-fold cross-validation [2].

[10] Using the training data, starting from 0 and incrementally, different values were tested from which 0.3 was finally selected for obtaining the best ERDE_{50}.

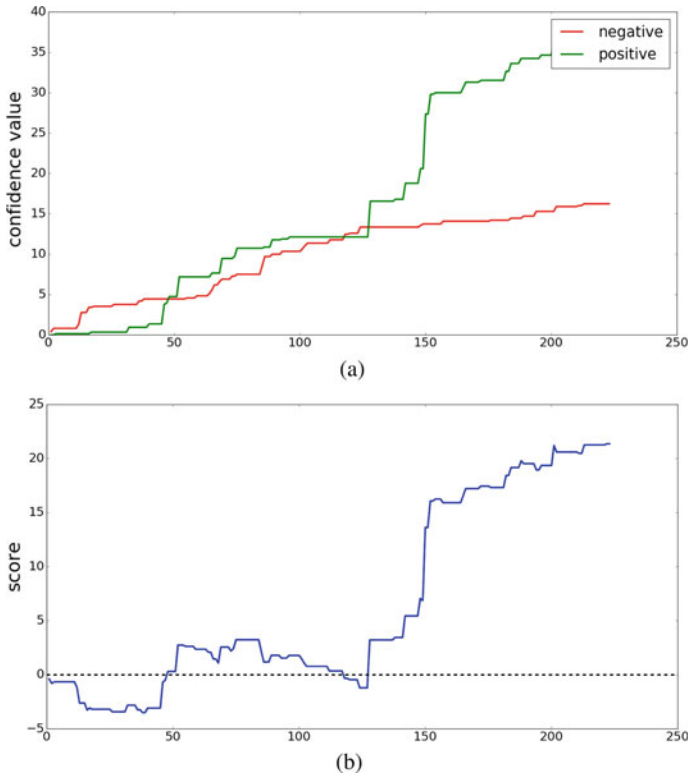

Fig. 4 Values for one of the actual users (subject 968) from the test set used for this task. **a** Variation of our model's positive and negative *confidence value* over time (post by post) for this user. **b** Variation of our model's estimated score over time for this user. Note that this user was classified as at-risk after processing around 50 of his/her posts, i.e. as positive as soon as the positive value exceeded the negative one or, equivalently, when the score became a positive number

As illustrated with an example in Fig. 4, this simple subtraction allows the two positive and negative confidence values to be represented, at any point in time, by a single and unified confidence value that no longer represents the confidence value for each class, but rather, a "decision-making" confidence value that represents the risk level estimated by the model used to classify and detect at-risk users.

Finally, after training our models we proceeded to the evaluation stage along with all the other 45 participant models, in which models had to connect to the remote server to receive user posts and to send their decisions along with the estimated risk level. The main results obtained with our 5 models are described below, grouped according to the type of metric used to measure performance:

Table 2 Results obtained for the early classification decision-based performance measures (sorted by $ERDE_{50}$). For comparison, the models that obtained the best results for the other 12 participating teams are also included

Model	$ERDE_5$ (%)	$ERDE^{\blacktriangle}_{50}$ (%)	$F_{latency}$	R	P	F_1
UNSL#4	6.14	**2.96**	0.46	0.92	0.31	0.47
UNSL#2	5.56	3.34	0.50	0.86	0.36	0.51
UNSL#3	5.59	3.48	0.49	0.85	0.35	0.50
UNSL#0	**5.53**	3.92	0.55	0.78	0.42	0.55
UNSL#1	5.68	4.10	0.55	0.75	0.43	0.55
CLaC#1	5.73	3.12	0.69	0.82	0.61	0.70
BiTeM#1	5.89	3.40	0.54	0.70	0.44	0.54
CLaC#4	6.25	3.42	**0.69**	0.79	0.64	**0.71**
UDE#0	8.48	3.87	0.58	0.74	0.51	0.61
UDE#1	7.48	3.94	0.53	0.73	0.44	0.55
INAOE-CIMAT#0	9.29	3.98	0.62	0.78	0.56	0.66
LTL-INAOE#1	7.74	4.19	0.55	0.75	0.47	0.58
INAOE-CIMAT#3	9.17	4.75	0.63	0.68	0.67	0.68
SINAI#2	9.04	4.89	0.30	0.95	0.18	0.30
INAOE-CIMAT#4	9.12	5.07	0.61	0.63	0.69	0.66
lirmm#0	9.13	5.14	0.63	0.63	0.74	0.68
lirmm#1	9.10	5.51	0.62	0.60	**0.77**	0.68
UppsalaNLP#4	5.73	5.66	0.41	0.42	0.40	0.41
BioInfo@UAVR#0	5.84	5.77	0.37	0.44	0.32	0.37
lirmm#3	9.08	6.62	0.48	0.42	0.74	0.54
SSN-NLP#3	7.61	6.86	0.33	0.26	0.48	0.34
HULAT#0	10.84	8.14	0.16	0.30	0.11	0.17
UDE#2	12.52	8.21	0.53	0.68	0.13	0.22
Fazl#2	17.11	11.22	0.14	**1**	0.09	0.16
Fazl#1	17.11	13.91	0.11	**1**	0.09	0.16

- **Early classification decision-based performance**: Table 2 shows the results obtained for the decision-based performance metrics. It can be observed that UNSL#0 obtained the best $ERDE_5$ and UNSL#4 the best $ERDE_{50}$ among all participating models. However, among our models, UNSL#4 was also the model with the lowest $F_{latency}$, which suggest that ignoring the words with low *global value* may contribute to improving performance in terms of the ERDE measure but at the cost of a worse performance in terms of the $F_{latency}$ measure.[11] The models

[11] This is probably caused by the final model heavily prioritizing recall over precision, affecting their harmonic mean which ultimately affected the $F_{latency}$. For instance, among our 5 models, UNSL#4 obtained the best recall (0.92) but the lowest precision value (0.31).

UNSL#2 and #3, trained with the entire training set, obtained a better performance in terms of the $ERDE_{50}$ measure compared to their two counterparts, UNSL#0 and #1, trained only with the training set corresponding to the previous edition. However, the performance of the former in terms of the $F_{latency}$ measure was lower compared to the latter, this is probably due to the fact that the selected hyperparameter configuration was meant to fit the model to the training data according to the $ERDE_{50}$ measure, not the $F_{latency}$—the $F_{latency}$ measure is based on the standard F_1 measure, as such, it favors models that assign equal importance to both recall and precision, whereas the ERDE measure models that prioritize recall over precision.[12] Unlike the results previously reported in the literature [4], in this particular task the use of word bigrams led to a slight loss of performance since the performance of the two models using word n-grams (UNSL#1 and #3) was slightly lower than their counterparts without n-grams (respectively, UNSL#0 and #2). Perhaps the fact that here, unlike in the reported results, users were processed one post at a time and not using a chunk-based approach—where even the most "hasty" models had to process at least one entire chunk before being able to make a decision. Another cause could be the fact that n-grams, although having superior semantic qualities, also suffer from inferior statistical qualities[13] which, depending on the data being used, could negatively affect performance. Finally, regarding the $F_{latency}$ measure, we did not achieve the best performance. More precisely, our best $F_{latency}$ values, obtained by UNSL#0 and #1, were 0.14 points below the best, 0.69, obtained by CLaC#4. This was mainly due to the fact that, as we mentioned above, our 5 models used the same hyperparameter configuration, originally selected to optimize the ERDE measure. Nevertheless, our best $F_{latency}$ value (0.55) was considerably above the average (0.38), and ranked 11th among the 50 participating models.

- **Performance in terms of execution time**: Table 3 shows, for each team, details on the total time used to complete the task. As can be seen, the time taken to complete the task differs widely from team to team, varying from a few to a large number of hours—one team even took as long as about a month. However, to have a more precise view of how efficient the models of each team were, not only the total time taken to complete the task must be considered, but also the total number of posts processed in that time, and the number of models used to carry it out. For example, in terms of processing speed, BiTeM does not seem as efficient as UNSL, since although the former completed the task in only 4 h, it only processed the first 11 posts from each user, while the latter, although completing the task in 23 h, it processed all 2000 posts from each user. Likewise, although BioInfo@UAVR took 14 h to classify all users with a single model, UNSL had to do it with 5 of them, which required not only 5 times more processing load but also

[12] The $ERDE_o$ measure is calculated with the cost of false positives (*cfp*) being considerably lower than that of false negatives (*cfn*). Note that giving more importance to recall than to precision is reasonable since, in early risk detection tasks, every single undetected (positive) user is a life at risk.

[13] The probability with which a particular sequence of words occurs will never be greater than the probability of each individual word.

Table 3 Details of the participating teams: team name, number of models (#*models*), number of user posts processed (#*posts*), time taken to complete the task and to process each post ($\frac{total\ time}{\#posts \times \#models}$)

			Time	
Team	#models	#posts	Total	Per post
UNSL⋆	5	2000	23h	**8s**
UppsalaNLP	5	2000	2 days + 7h	20s
BioInfo@UAVR	1	2000	14h	25s
UDE	3	2000	5 days + 3h	1m + 12s
lirmm	5	2024	8 days + 15h	1m + 12s
INAOE-CIMAT	4	2000	8 days + 2h	1m + 30s
HULAT	5	83	18h	2m + 36s
Fazl	3	2001	21 days + 15h	5m + 12s
BiTeM	4	11	4h	5m + 30s
LTL-INAOE	2	2001	17 days + 23h	6m + 30s
SINAI	3	317	10 days + 7h	15m + 36s
CLaC	5	109	11 days + 16h	31m
SSN-NLP	5	9	6 days + 22h	3h + 42m

5 times more requests sent to the server.[14] For this reason, Table 3 also includes an estimate of the time taken by each team's model to process each post, which was obtained by normalizing the total time relative to the number of models used and the total number of posts processed. Using this normalized time, it can be seen that UNSL models were the fastest, having processed each post in approximately 8 seconds[15] each—in other words, 2000 posts from each user processed with 5 models in 23 h. It is worth mentioning that our models were the fastest not because of superior computing capabilities,[16] but because the SS3 model was designed to be able to work incrementally, processing the input sequence in $O(n)$ time with respect to n, the length of the sequence [2]. This contrasts with other teams, such as CLaC [16], INAOE-CIMAT [1] or lirmm [17], which, although performed better in terms of the $F_{latency}$ measure, were considerably slower, taking them days to complete the same task. For example, the 5 CLaC [16] models were SVM

[14] For each of the users 2000 posts, not only was it necessary to send a request to the server to obtain the post, but also 5 more requests to send the response of each model. Therefore, for teams with 5 models like UNSL, completing the task required sending a total of $2000 + 2000 * 5 = 12000$ requests to the server. Therefore, if the connection latency was, for instance, 3 s, approximately 10 h of the total time would be consumed only by communication.

[15] Much of these 8 s corresponded to communication latency, since they include the latency of receiving the post, processing time, and the latency of sending the response.

[16] We coded our script in plain Python 2.7 and only using built-in functions and data structures; no external library was used (such as *NumPy*). Additionally, to run our script we used one of the author's laptop which had standard technical specifications (Intel Core i5, 8GB of DDR4 RAM, etc.).

classifiers that used neural features as input. More precisely, these features were extract from an ensemble approach that employs several attention-based neural sub-models, one such models (CLaC#1) obtained the best $F_{latency}$ value among all participating models. Nevertheless, these models were 232 times slower than UNSL's simple SS3 models, since CLaC took 11 days and 16 h to complete the task, having processed only 109 posts in total—which indicates that these models were computationally/algorithmically less efficient.

- **Ranking-based performance**: Table 4 shows the results obtained for the ranking-based performance metrics. It can be seen that our 5 models obtained, in the 4 rankings, the best values for the $P@10$ and $NDCG@10$ measures. Additionally, we also obtained the best $NDCG@100$ values for the rankings obtained after processing 1 and 100 posts, and the second and third best ones, respectively, after processing 500 and 1000 posts. Unlike with the decision-based metrics, here there was no noticeable difference in performance between the two SS3 models using word n-grams (UNSL#1 and #3) and their counterparts (respectively, UNSL#0 and #2). Between the models trained with the entire training set (UNSL#2, #3 and #4) and those trained only with the training set corresponding to the previous edition (UNSL#0 and #1), there was also no noticeable difference in terms of the $P@10$ and $NDCG@10$ measures, however, the former were considerably better in terms of the $NDCG@100$ measure. In other words, the three models that were trained with all available data, as expected, improved their ability to estimate the risk level of users—for example, the top-100 ranking of users ordered by the score given by UNSL#0, after processing the first 100 posts, was 77% perfect[17] (i.e. $NDCG@100 = 0.77$), whereas the one obtained by its counterpart, UNSL#2, was 83% perfect (i.e. $NDCG@100 = 0.83$).

	(after 500 post)			(after 1000 post)		
Model	P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100
UNSL#0	**1**	**1**	0.79	**1**	**1**	0.79
UNSL#1	**1**	**1**	0.79	**1**	**1**	0.79
UNSL#2	**1**	**1**	0.83	**1**	**1**	*0.84*
UNSL#3	**1**	**1**	0.84	**1**	**1**	*0.84*
UNSL#4	**1**	**1**	*.85*	0.9	0.94	*0.84*
UDE#1	**1**	**1**	**0.87**	**1**	**1**	**0.88**
UDE#0	0.9	0.93	*.85*	0.9	0.94	0.86
LTL-INAOE#0	0.9	0.92	0.73	0.7	0.78	0.65
Fazl#1	0.7	0.78	0.67	0.7	0.78	0.68
UppsalaNLP#4	0.8	0.75	0.52	0.8	0.75	0.52
BioInfo@UAVR#0	0.6	0.59	0.47	0.6	0.59	0.47

In general terms, it can be observed that, among all the participating models, the risk level estimated by our five SS3 models were the most consistent throughout the 4 rankings, having achieved the best values even when the ranking was generated after reading only the first post of each user. For example, it is interesting to note that, regardless of whether they were trained with the entire training set or not, or whether word n-grams were taken into account or not, all SS3 models were

[17] In this task, a perfect ranking is a ranking where all 73 at-risk users are located in the first 73 positions.

Table 4 Ranking-based evaluation results for the 4 reported rankings, respectively, the ranking obtained after processing 1, 100, 500, and 1000 posts. For comparison, the models that obtained the best results from the other 12 participating teams are also included

Model	(After 1 post)			(After 100 posts)		
	P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100
UNSL#0	**0.8**	***0.82***	0.54	**1**	**1**	0.77
UNSL#1	**0.8**	***0.82***	0.54	**1**	**1**	0.77
UNSL#2	**0.8**	***0.82***	***0.55***	**1**	**1**	0.83
UNSL#3	**0.8**	***0.82***	0.53	**1**	**1**	0.83
UNSL#4	**0.8**	***0.82***	0.52	0.9	0.94	***0.85***
LTL-INAOE#0	**0.8**	0.75	0.34	**1**	**1**	0.76
UDE#1	0.6	0.75	0.54	0.9	0.94	0.81
UppsalaNLP#4	0.8	0.75	0.52	0.8	0.75	0.52
BiTeM#1	0.8	0.75	0.47	–	–	–
SSN-NLP#3	0.6	0.64	0.3	–	–	–
BioInfo@UAVR#0	0.6	0.59	0.47	0.6	0.59	0.47
BiTeM#0	0.6	0.44	0.52	–	–	–
HULAT#0	0.3	0.33	0.18	–	–	–
Fazl#1	0.3	0.29	0.26	0.6	0.6	0.59
UDE#0	0.2	0.12	0.11	0.9	0.92	0.81
SINAI#0	0.2	0.12	0.11	–	–	–
CLaC#0	0.1	0.1	0.05	0.8	0.86	0.28
CLaC#1	0.1	0.1	0.04	0.3	0.45	0.16

able to construct a top-10 ranking of at-risk users which was 82% perfect (i.e. $NDCG@10 = 0.82$) after processing only the first post from each user. This indicates that our models, despite their simplicity, are capable of estimating the risk level of users with considerable efficiency, even when only a few posts have been processed.[18] This contrasts with other more complex models that, although performed better in terms of the $F_{latency}$ measure, were not as efficient in estimating risk levels—even when some took days to complete the task. For instance, although CLaC#1 obtained the best $F_{latency}$ value, ranking-based results show that its ability to estimate the risk level of the users was the lowest among all the participating models. Finally, it is important to mention that the ranking-based evaluation results obtained by our models imply two relevant points: (a) the score generated by SS3 (see Eq. 6) is estimating the risk level of the subjects correctly, and (b) the fact that user risk level is being estimated correctly implies that our models have room for improvement in terms of decision-based metrics (ERDE, $F_{latency}$, precision, and recall) by using better early classification policies.[19] This last point is not

[18] Which would explain why our models obtained the best ERDE values despite having classified all at-risk users, on average, after having processed only their first 2 posts.

[19] That is, if our models were not able to obtain better classification results, it was not due to a poor estimation of the risk level of the users, but due to the policy used to decide *when* to classify them based on that estimation.

Fig. 5 Word cloud of the top-100 words selected by *global value* learned by the SS3 model (UNSL#2) from the training set available for this task (words are sized by actual value)

minor, since it allows identifying possible lines of work to extend and improve our models, as will be discussed in the next and final section. Note that point (a) indicates us that the *global value* learned by the model, given by Eq. 1, is correctly capturing the degree of importance of each word—which could also be asserted from a more qualitative point of view in Fig. 5.

4 Conclusion and Future Work

In this chapter, we described the participation of our research team in the eRisk addressing the two editions of the early anorexia detection task. In the first edition, we used two main approaches, one based on a time-aware chunk-based document representation, called FTVT, and the other, on a novel text classifier that was specially designed for early risk detection scenarios, called SS3. In the second edition, given that evaluation data was released more realistically as a stream of (user) posts and not using the chunk-based approach, FTVT was no longer appropriate and, consequently, we addressed the task by focusing only on approaches based on the SS3 classifier.

Regarding the results obtained after the evaluation stage, in the first edition, we achieved the best $ERDE_5$ and the best precision values among all participant models, respectively, with one of our FTVT-based model and one of our SS3 models. It was interesting to note that the models using the FTVT representation performed well in terms of ERDE measures, even although the hyperparameter configuration, such as the FTVT n parameter, was not specially selected for this task (but rather for the

depression task). We also noticed that the FTVT model with no temporal enrichment of the positive class, i.e. the FTVT model with $n = 0$ equivalent to a standard CSA representation, obtained the lowest performance among all the participating models which strongly suggested that the "temporal enrichment" performed by the FTVT representation has, in fact, a considerable positive impact on the final performance of the classifiers. Also, regarding the SS3 model used for this edition, it is worth noticing that the performance in terms of standard (timeless) measures were comparable to that of the more complex and elaborate models such as the FHDO-BCSG CNN models pre-trained with 1.7 billion Reddit comments [21]. More precisely, our SS3 model (with $\lambda = 3$), which consisted simply of a summation of word values (Eq. 5) learned only from the limited available training data, not only achieved the best precision (0.91) but also obtained an F_1 value (0.79) that was only outperformed by those FHDO-BCSG models (0.81 and 0.85). Moreover, this SS3 model could also have obtained the best performance in terms of the ERDE measures if we had used a better hyperparameter configuration, as we later discovered [4]—which was the same configuration we used in the second edition. Finally, regarding the results obtained in the second edition, using SS3 with such hyperparameter configuration ($\lambda = \rho = 1$ and $\sigma = 0.45$), we were able to achieve the best values for both ERDE_5 and ERDE_{50} showing that this hyperparameter setting is robust in terms of this measure. Moreover, our models were the fastest in processing time and achieved remarkable results in terms of the ranking-based metrics, obtaining the best $P@10$ and $NDCG@10$ values, consistently, in all four rankings, and the best $NDCG@100$ values for the rankings after processing 1 and 100 posts—and the second and third best $NDCG@100$, respectively, for the rankings after 500 and 1000 posts. In general terms, the results suggest that, despite its simplicity, the SS3 classifier could be considerably competent when dealing with these types of scenarios. Results also suggest that this classifier possess certain robustness since, having used the same hyperparameter configuration and regardless of the data used to train it or whether word n-grams were taken into account or not, the performance of all five SS3 models was consistent across the three different dimensions used to measure it: execution time, users risk level estimation, and early classification performance.

Finally, for future work, we plan to explore better hyperparameter configurations to improve the performance of our models in terms of the $F_{latency}$ measure—for which we were unable to obtain the best result. More importantly, based on the promising ranking-based evaluation results suggesting that user risk level were properly estimated, we will also explore and design better early classification policies since the simple policy that we used proved to be "too hasty"—as we described earlier, all at-risk users were classified, on average, after having processed only their first 2 posts. For example, we believe that a more elaborated policy able to delay the decision until the estimated risk level is high enough, or the use of global information such as taking into account the current estimated risk level of all users to make the decision, could greatly improve the early classification performance.

References

1. Aragón, M. E., López-Monroy, A. P.,& Montes-y Gómez, M. (2019). INAOE-CIMAT at eRisk 2019: Detecting signs of anorexia using fine-grained emotions. In *Working Notes of CLEF, CEUR Workshop Proceedings*, Lugano, Switzerland.
2. Burdisso, S. G., Errecalde, M., and Montes-y Gómez, M. A text classification framework for simple and effective early depression detection over social media streams. Expert Systems with Applications 133 (2019), 182–197.
3. Burdisso, S. G., Errecalde, M., & Montes-y Gómez, M. (2019). UNSL at eRisk 2019: a unified approach for anorexia, self-harm and depression detection in social media. In *Working Notes of CLEF 2019, CEUR Workshop Proceedings*, Lugano, Switzerland.
4. Burdisso, S. G., Errecalde, M., & Montes-y Gómez, M. (2020). τ-SS3: A text classifier with dynamic n-grams for early risk detection over text streams. *Pattern Recognition Letters, 138,* 130–137.
5. Errecalde, M. L., Villegas, M. P., Funez, D. G., Ucelay, M. J. G., & Cagnina, L. C. (2017). Temporal variation of terms as concept space for early risk prediction. In *Working Notes of CLEF 2018, CEUR Workshop Proceedings*, Dublin, Ireland.
6. Funez, D. G., Ucelay, M. J. G., Villegas, M. P., Burdisso, S. G., Cagnina, L. C., Montes-y Gómez, M., & Errecalde, M. L. (2018). UNSL's participation at eRisk 2018 lab. In *Working Notes of CLEF 2018, CEUR Workshop Proceedings*, Avignon, France.
7. Gonzalez, A., Clarke, S., & Kohn, M. (2007). Eating disorders in adolescents. *Australian Family Physician, 36,* 8.
8. Hay, P. (2020). Current approach to eating disorders: a clinical update. *Internal Medicine Journal, 50,* 24–29.
9. Kakhi, S., and McCann, J. Anorexia nervosa: diagnosis, risk factors and evidence-based treatments. Progress in Neurology and Psychiatry 20 (2016), 24–29.
10. Li, Z., Xiong, Z., Zhang, Y., Liu, C., and Li, K. Fast text categorization using concise semantic analysis. Pattern Recognition Letters 32, 3 (2011), 441–448.
11. López-Monroy, A. P., Montes-y Gómez, M., Escalante, H. J., Villasenor-Pineda, L., and Stamatatos, E. Discriminative subprofile-specific representations for author profiling in social media. Knowledge-Based Systems 89 (2015), 134–147.
12. Losada, D. E., & Crestani, F. (2016). A test collection for research on depression and language use. In N. Fuhr, P. Quaresma, T. Gonçalves, B. Larsen, K. Balog, C. Macdonald, L. Cappellato & N. Ferro (Eds.),*Experimental IR Meets Multilinguality, Multimodality, and Interaction* (Cham, 2016) (pp. 28–39). Springer International Publishing.
13. Losada, D. E., Crestani, F., & Parapar, J. (2017). eRisk 2017: CLEF lab on early risk prediction on the internet: Experimental foundations. In G. J. Jones, S. Lawless, J. Gonzalo, L. Kelly, L. Goeuriot, T. Mandl, L. Cappellato, & N. Ferro, (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction* (Cham, 2017) (pp. 346–360). Springer International Publishing.
14. Losada, D. E., Crestani, F., & Parapar, J. (2018). Overview of eRisk: Early risk prediction on the internet. In P. Bellot, C. Trabelsi, J. Mothe, F. Murtagh, J. Y. Nie, L. Soulier, E. SanJuan, L. Cappellato & N. Ferro, (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction* (Cham, 2018) (pp. 343–361). Springer International Publishing.
15. Losada, D. E., Crestani, F., and Parapar, J. (2019). Overview of eRisk 2019 early risk prediction on the internet. In F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D. E. Losada, G. Heinatz Bürki, L. Cappellato & N. Ferro (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction* (Cham, 2019) (pp. 340–357). Springer International Publishing.
16. Mohammadi, E., Amini, H., & Kosseim, L. (2019). Quick and (maybe not so) easy detection of anorexia in social media posts. In *Working Notes of CLEF 2019, CEUR Workshop Proceedings*, Lugano, Switzerland.
17. Ragheb, W., Azé, J., Bringay, S., & Servajean, M. (2019). Attentive multi-stage learning for early risk detection of signs of anorexia and self-harm on social media. In *Working Notes of CLEF 2019, CEUR Workshop Proceedings*, Lugano, Switzerland.

18. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1, 5 (2019), 206–215.
19. Sadeque, F., Xu, D., & Bethard, S. (2018). Measuring the latency of depression detection in social media. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (pp. 495–503).
20. Termorshuizen, J., Watson, H., & Thornton, LM, et al. (2020). Early impact of COVID-19 on individuals with self-reported eating disorders: A survey of 1,000 individuals in the United States and the Netherlands. *International Journal of Eating Disorders, 53*, 1780–1790.
21. Trotzek, M., Koitka, S., & Friedrich, C. M. (2018). Word embeddings and linguistic metadata at the CLEF 2018 tasks for early detection of depression and anorexia. In *Working Notes of CLEF 2018, CEUR Workshop Proceedings*, Avignon, France.
22. Vikram, P. (2005). *Gender in Mental Health Research*. Gender and Health Research Series: World Health Organization.
23. Zhang, Y., Jin, R., & Zhou, Z.-H. (2010). Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics, 1*(1–4), 43–52.