ETM: Enrichment by Topic Modeling for Automated Clinical Short Text Classification to Detect Patients' Disease History

Ayoub Bagheri[a,b], *Arjan Sammani*[b], *Peter G.M. van der Heijden*[a,c], *Folkert W. Asselbergs*[b,d,e], and *Daniel L. Oberski*[a,f]

Email: *a.bagheri@uu.nl; a.bagheri-2@umcutrecht.nl*

[a]Department of Methodology and Statistics, Utrecht University, Utrecht, The Netherlands

[b]Department of Cardiology, Division of Heart and Lungs, University Medical Center Utrecht, Utrecht, The Netherlands

[c]S3RI, Faculty of Social Sciences, University of Southampton, Southampton, UK

[d]Institute of Cardiovascular Science, Faculty of Population Health Sciences, University College London, London, UK

[e]Health Data Research UK, Institute of Health Informatics, University College London, London, UK

[f]Julius Center for Health Sciences and Primary Care, University Medical Center Utrecht, Utrecht, The Netherlands

Abstract

Given the rapid rate at which text data are being digitally gathered in the medical domain, there is growing need for automated tools that can analyze clinical notes and classify their sentences in electronic health records (EHRs). This study uses EHR texts to detect patients' disease history from clinical sentences. However, in EHRs, sentences are less topic-focused and shorter than that in general domain, which leads to the sparsity of co-occurrence patterns and the lack of semantic features. To tackle this challenge, current approaches for clinical short text classification are dependent on external information to improve classification performance. However, this is implausible owing to a lack of universal medical dictionaries. This study proposes the ETM (enrichment by topic modeling) algorithm, based on latent Dirichlet allocation, to smoothen the semantic representations of short texts. The ETM enriches text representation by incorporating probability distributions generated by an unsupervised algorithm into it. It considers the length of the original texts to enhance representation by using an internal knowledge acquisition procedure. When it comes to clinical predictive modeling, interpretability improves the acceptance of the model. Thus, for clinical short text classification, the ETM approach employs an initial TFiDF (term frequency inverse document frequency) representation, where we use the support vector machine and neural network algorithms for the classification task. We conducted three sets of experiments on a dataset consisting of clinical cardiovascular notes from the Netherlands to test the sentence classification performance of the proposed method in comparison with prevalent approaches. The results show that the proposed ETM approach outperformed state-of-the-art baselines.

Keywords: short text classification . sentence classification . clinical text classification . latent Dirichlet allocation . enriched text representation

1. Introduction

In recent years, with the development of intelligent information systems for electronic health records (EHRs) inferring patterns, topics, and knowledge from large-scale clinical textual data has emerged as an important and challenging task for a wide range of healthcare applications, such as the classification of disease history, event prediction, topic detection, and patient identity anonymization. While using free text in EHR is useful for medical practitioners, it poses technical challenges for text mining and natural language processing (NLP) [1, 2, 3, 4]. Some challenges in this area are short sentences, inconsistent structure between texts, unstructured texts, abbreviations, and errors of spelling and grammar. In light of the above, there is a need for tools for automatic text mining to extract implicit, previously unknown, and useful information from data. This study proposes a text mining model for patients' disease history detection, where the records are sentences and labels are binary values that show the presence of disease history.

Many researchers have examined the task of mining clinical text for applications in healthcare [1, 2, 5, 6, 7, 8, 9, 10] and have approached it as a basic text classification problem. Two major challenges in clinical text classification are the unstructured and short representations of text. Short texts refer to texts with limited context, where the sparsity of patterns of word co-occurrence in the content makes text mining difficult [11, 12, 13, 14, 15, 16]. The very small number of words in one medical sentence in EHR texts leads to a large classification error [11, 16].

In clinical text mining, the problem of short text classification is often disregarded [17]. Studies on short text classification for EHR data are mainly based on external dictionaries (ontologies) created by medical experts. In practice, we often do not have dictionaries or do not know in advance of ontologies that might be relevant to the specific domain of application. In addition, for clinical prediction, model interpretability helps to understand the distribution of outcome based on the input words. Therefore, there is increasing demand for automated explainable tools that can analyze and classify EHR free texts. In this study, with the aim of extracting sentences containing medical history from EHR texts, we propose the ETM (enrichment by topic modeling) algorithm for automatic sentence classification for clinical notes. The novelty of the ETM is in the underlying clustering approach that extracts related knowledge from the dataset without the need for external dictionaries, such as a medical ontology, to tackle the sparsity of patterns of word co-occurrence. The proposed clustering algorithm is based on latent Dirichlet allocation (LDA) algorithm [18], and uses a dynamic weighting mechanism to enrich the data. This algorithm first clusters the initial dataset of clinical notes to generate the distribution of hidden topics in clinical notes and probabilities of words in the given topic. Subsequently, the proposed weighting mechanism assigns a weight to every text in terms of its length to mitigate the sampling error inherent in sparse texts by interpolating between the observed word counts and the implied number obtained from an unsupervised model. The proposed ETM yields a smoothed dataset that balances individual observations with generic patterns extracted by the LDA algorithm to improve short text classification.

This study uses clinical notes from a dataset collected by the department of Cardiology of the University Medical Center Utrecht (UMCU). The UMCU EHRs encompass free text fields in which different short clinical snippets can be entered: e.g. patient anamnesis, physical examination, and medical history. Patients' disease history detection is an example of one classification task on short text from UMCU clinical notes. In this study, each short text is considered one sentence of about 12 words. Medical personnel at the department of Cardiology of the UMCU have requested such a system to help them understand past cases from EHR records with similar histories to present ones. Given the nature of free texts, short text (sentence) classification is necessary as the first step to extract the disease history, where not all medical history is clearly delineated and may be provided in free texts at the discretion of the physician.

Thus, this study contributes to the field in the following ways: (i) It presents a method for automatic short text classification for clinical notes to tackle the problem of the sparsity of patterns of word co-occurrence. (ii) It uses the output of the clustering algorithm as an internal source for enriching short texts. (iii) It uses the composition of the topic–word distributions and topic distributions of a document with the interpretable TFiDF (term frequency–inverse document frequency) representation. (iv) It takes the shortness of the text into account for enriched representation.

The remainder of this paper is structured as follows: Section 2 gives an overview of related work on clinical text classification and short text classification. Section 3 presents an intuitive explanation of the proposed unsupervised model-based smoothing idea. In Section 4, we introduce the proposed ETM approach, and Section 5 details experimental evaluations of the proposed method and a discussion of the results. It shows the usefulness of the proposed method for clinical short text classification. Finally, in Section 6, we offer concluding remarks and directions for future research.

2. Related work

2.1. Clinical text classification

EHR data contain a large amount of text in which useful patterns need to be automatically identified. Machine learning and text mining algorithms with different data representation methods have been used to study the classification of clinical notes. Mujtaba et al. [19] presented a comprehensive review of articles on clinical text classification published in 2013–2018. Based on their study, the most extensively employed clinical texts for classification are pathology reports, radiology reports, and Medline biomedical documents. In a majority of studies, bag of words (BOW) representations: binary, term frequency, and TFiDF feature representations were determined to be beneficial. A significant number of the studies have used either supervised machine learning or rule-based approaches.

Many approaches to clinical text classification rely on medical ontologies (dictionaries), such as the unified medical language system (UMLS) meta-thesaurus, and medical subject headings (MeSH), to glean

knowledge from clinical notes. Yao et al. [20] proposed an approach that combines rule-based features and a knowledge-guided convolutional neural network for effective disease classification. They used concepts from the UMLS meta-thesaurus. Similarly, Kocbek et al. [21] combined three clinical reports—from pathology, radiology, and patients' admission-related meta-data—and used a support vector machine (SVM) with a bag of phrases from the UMLS meta-thesaurus to predict the rate of admissions against disease.

On the contrary, some clinical text classification studies have used non-dictionary-based approaches instead of dictionary-based methods. For instance, Bui and Zeng [22] applied regular expressions to extract snippets of text from clinical notes containing specific words and built an SVM classifier to categorize them. Fodeh et al. [23] used unstructured text narratives in the EHR to derive pain assessments from clinical notes on patients with chronic pain. They developed their system based on different machine learning classifiers, among which random forest achieved the best results. Blanco et al. [24] used several deep learning classification models for assigning multiple ICD codes to clinical documents. They implemented binary logistic regression, a neural network with three fully connected hidden layers, and a bidirectional gated recurrent unit for text classification.

Nevertheless, the problem of clinical short text classification was not covered in work by Mujtaba et al. [19], because a few studies have sought to derive the knowledge hidden in clinical short text [5, 17, 19, 25, 26]. Hughes et al. [25] applied convolutional neural networks (CNNs) with a distributed word representation to medical text classification at the sentence level. They evaluated the learning of complex data representations using the algorithm instead of feature engineering for clinical knowledge representation. Lv et al. [26] used sentence segmentation, word segmentation, part of speech and entity extraction for text preprocessing to extract features for short text classification in EHRs. In their approach, $TFiDF$ and latent semantic analysis are used to select features that represent the vocabulary for short text classification from several entity dictionaries. In addition, a dependency parser is applied to texts where the dependency relations are used as features for text classification. Cao et al. [17] proposed a knowledge-guided short text classification system for healthcare applications, and claimed that text in the healthcare domain contains domain-specific or infrequently appearing words that can lead to poor embedding owing to a lack of training data. They proposed a bidirectional long short-term memory deep neural network to perform short text classification tasks. Their approach is a domain knowledge-guided attention model that uses the domain dictionary at hand to refine classification performance.

The main difference between the above studies on clinical short text classification and our approach is that the former studies used domain dictionaries and disregarded the unlabeled data. Our approach uses the unlabeled data for the unsupervised model-based smoothing, and deploys the labeled data for the short text classification model.

2.2. Short text classification

Impressive progress has been made on the problem of text classification, but few studies have tackled short texts [9, 11, 13, 14, 27, 28, 29]. Unlike the traditional text classification problem, short texts pose two main challenges. First, patterns of word co-occurrence are sparse in the feature space, where a short text contains only several to a dozen words. For example, on Twitter or a simple text message on a phone, users are constrained to a 140-character limit that cannot provide enough frequency information compared with the dense and diverse feature representations in the classification of longer texts. Second, texts face the challenge of a large-scale and manual labeling task, where with short texts this task is more burdensome as they are very small samples causing to increase noise and reduce classification accuracy.

Several techniques have been proposed to tackle the challenges posed by short text classification, including dimension reduction [11, 12, 30], topic modeling [13, 31, 32], clustering [9, 14, 30, 33, 34, 35], and word embedding [9, 36, 37]. Zelikovitz and Hirsh [11] developed a method to reduce error rates in short text classification by using a combination of labeled training data plus a large body of "uncoordinated background knowledge" that is a secondary corpus of unlabeled but related longer documents. They used the WHIRL method [38] for text classification, an information integration tool designed to query and integrate varied sources of text from the Web. Sriram et al. [12] proposed an intuitive approach to classify the short texts in tweets by using author information and features of texts. Yin et al. [14] proposed a short text classification technique based on a combination of the K-nearest neighbors (KNN) and hierarchical SVM classification. They used KNN to initially group labels of the samples to create subclasses and then they applied a SVM algorithm as a hierarchical multi-class classification to each group to classify labels. Cheng et al. [13] proposed a biterm topic model to capture topics in short texts based on aggregated biterms in the entire corpus to tackle the sparsity of patterns of word co-occurrence in texts. They defined the biterm as an unordered word pair co-occurring in a short text. They considered the corpus as a mixture of topics, where each biterm is drawn independently from a specific topic. Yang et al. [32] proposed a topic model to extract key phrases for short text classification using the idea that knowledge incorporation can solve the problem of sparsity. Their approach extracts topics from texts by focusing on phrases in the generative process of documents.

Bollegala et al. [30] developed ClassiNet, a network of binary classifiers trained to predict missing features from a given short text for text classification. ClassiNets solves the problem of feature sparseness by generalizing word co-occurrence graphs by considering implicit co-occurrences between features. Dai et al. [33] proposed the Crest to generate topic clusters from training data by exploiting a clustering method. Crest uses topic information to extend the representation of short texts and define a new feature space. It subsequently measures the cosine similarity between a document and clusters as augmented features of the document for classification.

Lee et al. [36] presented a model on the basis of recurrent and convolutional neural networks. Their

model incorporates preceding short texts for sequential short text classification. This model comprises two parts. The first part generates a vector representation for each text and the second part classifies the vector representations of the current text as well as a few preceding short texts using a two-layer feed-forward neural network. Kozlowski and Rybinski [9, 34] used a neural network-based distributional model for enriching the semantic meaning of short texts for clustering. They proposed the SnSRC clustering algorithm that uses the SnS method [34], a knowledge-poor text mining algorithm to sense induction, a language-independent approach. They trained their model using continuous bag of words and negative sampling, and computed cosine similarity between the mean vector of the embeddings for the text and the vectors for each word in the distributional model. The retrieved words with the highest semantic similarity were added as additional term features to the initial BOW text representation. In their study, especially in cases involving a specific domain language, the semantic enrichment of texts by applying neural networks improved the quality of clustering. Hill et al. [37] overcame feature sparseness in sentence representations by embedding them into a low-dimensional, dense space. They compared deep neural language models that compute sentence representations from unlabeled data with prevalent methods for word representation, and concluded that the unsupervised BOW models delivered the best performance in terms of sentence representation compared with supervised ones.

Current methods for short text classification either represent texts in a lower-dimensional space to reduce feature sparseness or add data to the text to enhance the quality of the feature space. The main outstanding challenge is the construction of external knowledge repositories, a labor-intensive task in applications of domain-specific clinical text mining. We propose an approach to tackle this challenge in clinical short text classification that deploys an unsupervised scheme for enriching the original dataset by internal knowledge acquisition, where the length of each document is considered by a dynamic weighting mechanism. The proposed approach uses the output of the unsupervised scheme as an internal source for enriching that does not employ any external dictionary.

3. Intuitive explanation

The intuition behind using a clustering method is that, in the BOW representation, short texts are simply very small samples from an underlying multinomial distribution: in this situation, smoothing should present a favorable bias-variance tradeoff, particularly if the smoothing is done towards a latent representation correlated with the outcome. Figure 1 illustrates this intuition.

Panel A shows a highly simplified representation of documents as *hypothetical* coordinates in the simplex formed by the true proportion of the words "hypertension" and "complaint" in each document. A hypothetical decision boundary for a binary outcome is also shown. Panel B shows the effect of observing only short texts: each point is a sample from the binomial sampling distribution $\hat{\pi}_i \sim \mathcal{N}[\pi_i, \pi_i(1-\pi_i)/n]$, where the number of words is taken to be small, $n = 10$. The unobservable true points, π_i, are shown as gray crosses.

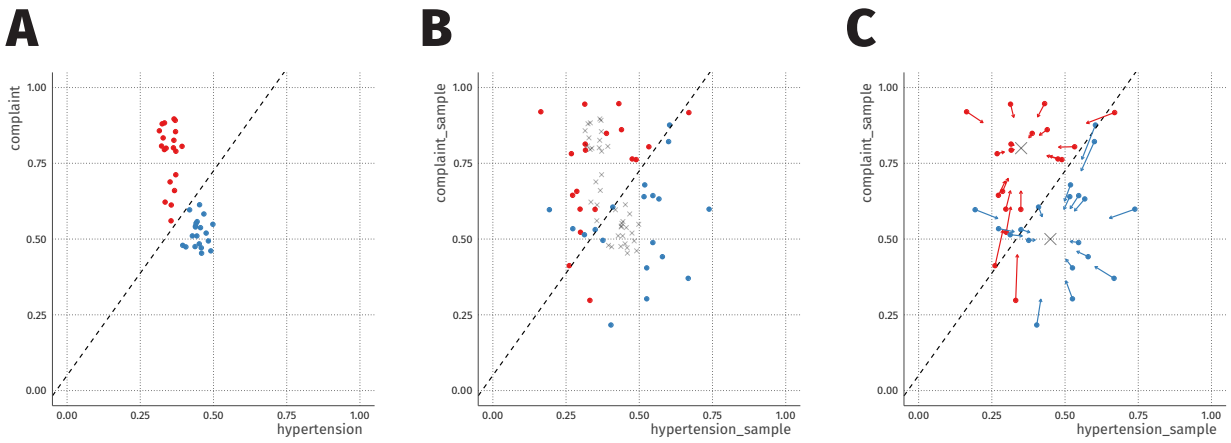

Figure 1: Intuition behind the proposed algorithm. Panel A: hypothetical documents, represented as coordinates in a simplex, separated by a decision boundary. Panel B: short texts are samples from the original simplex with small number of words ($n = 10$), increasing noise and reducing classification accuracy. Panel C: Clustering shrinks each observed coordinate to an estimated topic centroid, improving classification accuracy.

Due to the noise incurred from small sample size, many points are on the wrong side of the decision boundary. Panel C demonstrates the effect of using the proposed clustering-based algorithm, which consists of estimating topic centroids (crosses in panel C) and smoothing the observed coordinates towards these estimated centroids. For simplicity of illustration, here smoothing has been performed as $\tilde{\pi}_i = (1-\alpha)\hat{\pi}_i + \alpha\hat{\pi}^*_{k_i}$, where $\hat{\pi}^*_{k_i}$ is the coordinate of the centroid to which point i is estimated to belong, and the amount of smoothing is taken as $\alpha = 0.3$. As can be seen in Figure 1, the smoothing (1) reduces the variance of the estimates $\tilde{\pi}_i$, and (2) tends to take misclassified points back across the classification boundary, improving accuracy.

4. Proposed Methodology

The model for clinical short text classification proposed in this study is shown in Figure 2. This model consists of the following four steps.

- **Data representation,** i.e., preprocessing of clinical texts consisting of sentence detection and extraction, tokenization, spell correction, and representation.
- **LDA clustering,** i.e., using the LDA topic model to cluster short notes in collections of documents to obtain the probabilities of the distributions of document–topic and topic–word in the dataset.
- **ETM: topic-based smoothing,** i.e., using the ETM algorithm as a smoothing method to enrich the representation of clinical short notes according to distribution probabilities of the LDA model.
- **Classification,** i.e., using machine learning classifiers to classify each enriched short note. The classification algorithms used in this model are discussed in the experiments' section.

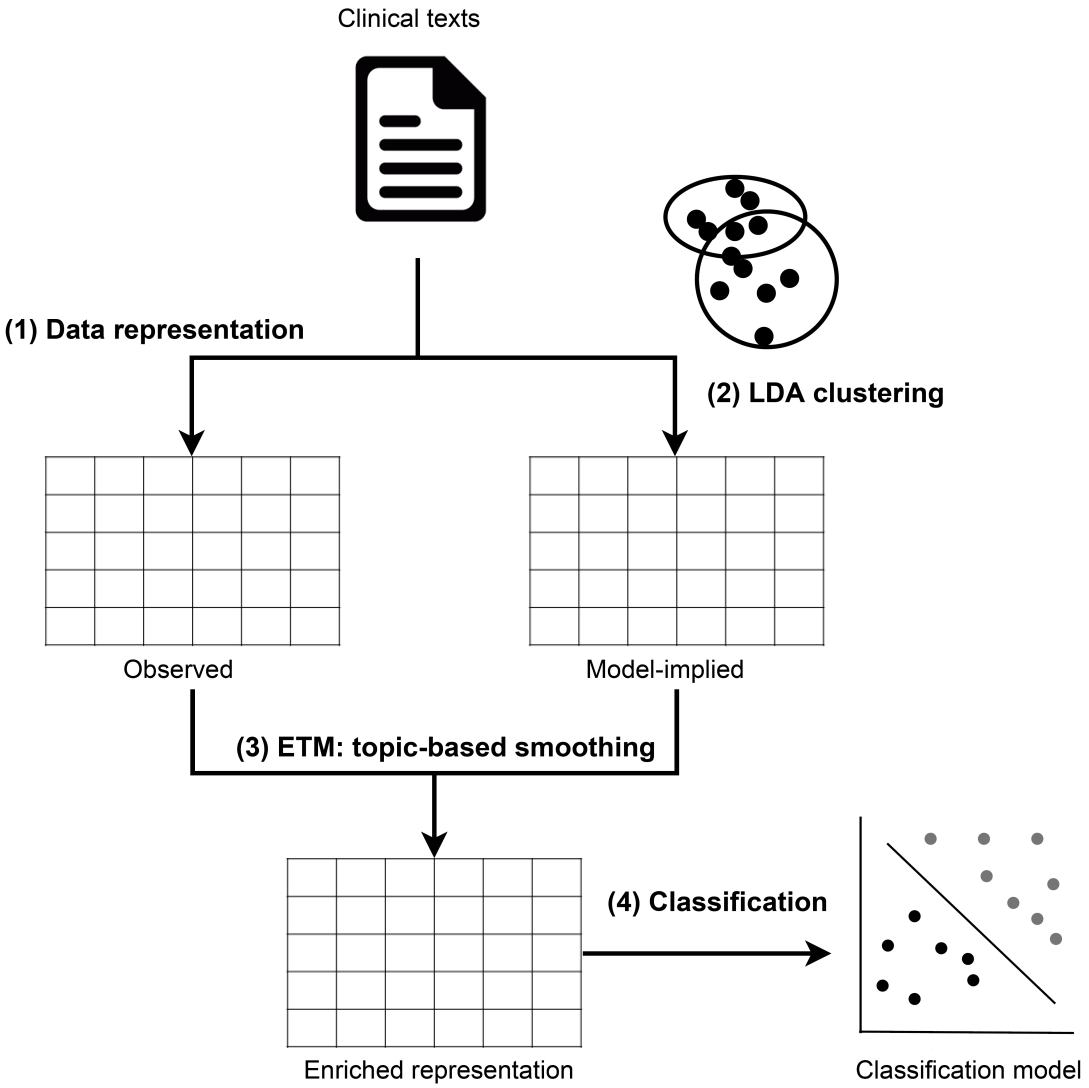

Figure 2: The clinical short text classification model.

4.1. Data representation

DEDUCE [39], a pattern matching tool, is used for automatic de-identification of Dutch medical texts, to anonymize clinical notes for legal and privacy reasons. All texts are tokenized using NLTK library [40] and the Python scikit-learn [41] feature extractor. NLTK sentence tokenizer uses an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences; and then uses that model to find sentence boundaries. This approach has been shown to work well for many European languages [40]. The punctuation marks that are commonly used to separate sentences in our case study are period, question mark, exclamation point, and semicolon. To handle spelling errors in Dutch texts, the Python

package language-check[1] is used, which is a wrapper for the LanguageTool[2] package. LanguageTool is an open source proofreading software that can detect and correct spelling errors in more than 20 different languages.

Each clinical short note (document) d from the dataset is represented by a normalized V-dimensional vector weighted by the TFiDF measure. TFiDF is a BOW representation model that stands for term frequency—inverse document frequency, and is defined as follows:

$$\begin{aligned} \mathrm{tf}_{d,i} &= \frac{n_{d,i}}{\sum_v n_{d,v}} \\ \mathrm{idf}_i &= \log \frac{|C|}{|C_i|} \\ \mathrm{tfidf}_{d,i} &= \mathrm{tf}_{d,i} \times \mathrm{idf}_i \end{aligned} \tag{1}$$

where V is the size of the vocabulary, $n_{d,i}$ denotes the number of times the ith word appears in document d, $|C|$ denotes the total number of documents in the dataset, and $|C_i|$ is the number of documents containing the ith word. TFiDF evaluates how important a word is to a document in a dataset, where the importance increases proportionally to the number of times a word appears in the document but is offset by the document frequency of the word in the dataset. Thus, with this representation, each document in the dataset can be regarded as a multinomial distribution over V words, and each dimension reflects the semantic coherence between the ith word and the document d.

4.2. LDA clustering

Topic modeling is a way of discovering topics in unlabeled text data [13, 18]. The LDA is a generative topic model that represents documents as a mixture of topics and assigns certain probabilities to the words. In other words, the LDA model is an unsupervised learning method that seeks patterns by inferring hidden variables in texts by treating words as observations.

Given a document in the form of $d = (w_1, w_2, ..., w_m)$ with a text dataset (corpus) D_n, and given N asked-for topics, the LDA model estimates the distribution of hidden topics in each document, known as parameter θ, and the probability of each word given the topic as β. Figure 3 shows a graphical representation of the LDA model [18], where the nodes are random variables and the edges indicate the conditional dependencies between them. The shaded and unshaded variables indicate observed and latent (i.e., unobserved, hidden) variables, respectively, while the plates refer to repetitions of the steps of sampling with the variable in the lower-right corner referring to the number of samples. As Figure 3 shows: the parameter α is a dataset-level Dirichlet prior that can be interpreted as the prior number of observations of a topic being sampled in a

[1] https://github.com/myint/language-check
[2] https://languagetool.org

document before having observed any words from the document. Similarly, parameter η is a dataset-level Dirichlet prior that can be interpreted as the number of prior observations of words sampled from a topic before any word from the dataset is observed. These two parameters are assumed to be sampled once in the LDA model when generating a dataset of documents.

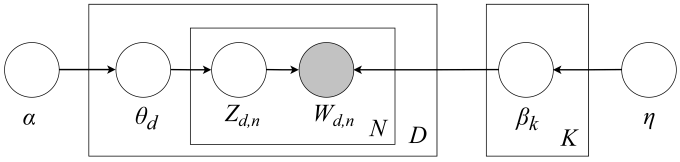

Figure 3: The graphical representation of the LDA model.

The variable θ_d is a document-level variable that is sampled once for each document. θ_d is the distribution of hidden topics in document d based on a multinomial distribution with the Dirichlet parameter α. The variable β_k is a topic-level variable that represents the probability distribution of words in topic k. Variables $Z_{d,n}$ and $W_{d,n}$ are word-level variables that are sampled once for each word in each document ($n \in \{1, 2, ..., N\}$). The variable $Z_{d,n}$ is a topic generated by a multinomial distribution with the parameter θ, and variable $W_{d,n}$ is a word sampled from the multinomial distribution with parameters β and Z.

The process of the LDA-clustering algorithm [18] implies a joint distribution over the latent and observed random variables (W, Z, β, θ) defined as follows:

$$p(W, Z, \beta, \theta|\alpha, \eta) = \prod_{k=1}^{K} p(\beta_k|\eta) \times \prod_{d=1}^{D} p(\theta_d|\alpha) \left(\prod_{n=1}^{N} p(Z_{d,n}|\theta_d) p(W_{d,n}|Z_{d,n}, \beta_{d,k}) \right) \tag{2}$$

Standard statistical techniques can be used to invert the generative process of the LDA model, thus inferring the set of topics responsible for generating a collection of documents. To use the LDA, the key inferential problem to solve is that of computing the posterior distribution in Equation 3 of the hidden random variables given the observed words in a document:

$$p(Z, \beta, \theta|W, \alpha, \eta) = \frac{p(W, Z, \beta, \theta|\alpha, \eta)}{p(W|\alpha, \eta)} \tag{3}$$

$$p(W|\alpha, \eta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n \in Z} p(z_n|\theta) p(w_n|z_n, \beta) p(\beta|\eta) \right) d\theta \tag{4}$$

This posterior distribution is intractable to compute, and thus approximate inference algorithms are needed for the posterior estimations of β, θ, and Z. The most common approaches used for making inferences in the LDA model are expectation maximization, Gibbs sampling, and variational inference.

4.3. ETM: Topic-based smoothing

Short text classification is different from traditional text classification in the brevity of the text involved. A simple solution to improve classification is to enrich the data representation of short texts before training a machine learning model.

Two main approaches have been used to enrich the representation of short texts. The first is to obtain the contextual information of short text and add more data, and the second approach involves uncovering latent topics from a dataset and adding topic-related information to smoothen the representation of short texts. We combine the ideas underlying these two approaches by introducing the ETM algorithm as a topic-based smoothing method. To enrich the feature space of a short text, the ETM algorithm matches the inferred probability distributions from the LDA model to words of short texts. Short texts are represented as a $TFiDF$ matrix, a $N \times V$ matrix where the rows denote the texts and the columns contain $TFiDF$ values of the chosen words. Then, by applying a method of inference, the ETM extracts the topic distributions and topic assignments for the $TFiDF$ matrix of texts.

The ETM exploits topic analysis to enhance features in short texts by assigning weights to words on the basis of the topics of inference, as the internal features of texts. This approach is applied to clinical notes to improve classification performance. When dealing with short texts, especially when manual labeling is labor intensive, the ETM can use unlabeled data to enrich the quality of the available labeled data. The overall procedure of the ETM is outlined in Algorithm 1.

Algorithm 1: ETM algorithm

Input: θ, β, M
Output: C
for *each document in* $d = \{1, 2, ..., D\}$ **do**
    Calculate γ_d by Equation 5;
    for *each word in* $i = \{1, 2, ..., N\}$ **do**
        $\omega = 0$
        for *each topic in* $k = \{1, 2, ..., K\}$ **do**
            $\omega = \omega + \theta_{d,k}\beta_{k,i}$
        end
        $C_{d,i} = M_{d,i} + \omega\gamma_d$
    end
end

In this algorithm, M is the $TFiDF$ matrix and C is the enriched matrix. D is the number of texts in the dataset, K is the number of topic clusters, and N is the number of words in the vocabulary. For each

short text, the ETM computes a dynamic enrichment weight γ as in Equation 5:

$$\gamma_d = \frac{m}{n_d} \tag{5}$$

m is the average length of the short texts and n_d is the number of words in the text document indexed by d. The weight γ is computed to consider the length of each short text in the enrichment with the ETM. This means that if a text is longer than the average, the weight of the enrichment decreases. The ETM calculates ω as in Equation 6, where ω is the enrichment value when information on the distributions θ and β is available. The ETM considers the original representations using β as the posterior probability of each word given the topic, and θ is the posterior distribution of hidden topics in each document. The ETM updates the representation by adding the enrichment value ω as in Equation 6.

$$\begin{aligned} \omega_{d,i} &= \sum_{d=1}^{D}\sum_{i=1}^{N}\sum_{k=1}^{K}\theta_{d,k}\beta_{k,i} \\ C_{d,i} &= M_{d,i} + \sum_{d=1}^{D}\sum_{i=1}^{N}\omega_{d,i}\gamma_d \end{aligned} \tag{6}$$

The ETM algorithm enriches each document of the dataset by incorporating the length of the document, the posterior distribution of hidden topics in the document, the probabilities of each word given the topic, and the value of the $TFiDF$ of the word. This algorithm considers the length of each document as it incorporates an enrichment dynamic weight with greater values for shorter documents. The idea of considering the length of a text in the enrichment process is as in empirical datasets, where some sentences are long enough while others are short and need to be enriched. The ETM assumes that each document is a mixture of corpus-wide topics, and gains internal knowledge by taking advantage of the patterns of clustering. These patterns contain more contextual information on short texts that can improve clinical short text classification performance. Because the ETM algorithm uses internal knowledge of the dataset as the main source of enrichment, its effectiveness is data dependent.

5. Evaluation experiment

In this section, we present the results of clinical short text classification using several classification algorithms. We evaluated the proposed approach from three aspects. First, we compared the ETM, using different numbers of topic clusters, with the original representations of short texts. Second, we ran experiments using unlabeled data. Third, we compared the ETM with two recently developed methods: Crest [33] from a short text classification study, and a CNN-based approach [25] from a medical text classification study.

Table 1: Dataset description

Category	Number of sentences	Average number of words
Labeled: medical history	3,560	16.51
Labeled: no medical history	7,493	8.76
Unlabeled	20,200	11.19

5.1. Data

The UMCU is one of the largest university hospitals in the Netherlands that provides specialized cardiac care. Given the structure of its EHRs, the data are available on a research data platform and can be extracted accordingly. The textual dataset used in this study consisted of all clinical cardiovascular notes from doctors or physicians' assistants between 2014 and 2018. A total of 1,002 clinical notes were manually annotated for medical history based on the International Classification of Diseases (ICD10)[3] criteria, and were checked sample wise by doctors. Conflicts in annotation were resolved through discussion. The words in the clinical notes on which the annotation was based were also marked for text mining. These words delineated sentences as records in our dataset describing medical history. The train dataset was generated based on this delineation. The description of the dataset is provided in Table 1. The train and unlabeled data contained 11,053 and 20,200 sentences, respectively. Sentences in the train data were labeled as two classes: with and without medical history. A total of 3,560 records had medical history and 7,493 records were labeled as without medical history.

5.2. Example

We present an example (Figure 4) to demonstrate the first three steps of the clinical short text classification model, in this study. This example is used to describe the idea of how text representation could be enriched by incorporating probability distributions from the LDA clustering algorithm.

Data provided in this example contains five sentences and 14 unique words. M is the initial BOW representation, the $TFiDF$ matrix and C is the output of the ETM algorithm, the enriched matrix. The LDA model was applied on the dataset to learn two clusters of words (topics). β represents the probability distribution of words for the topics T^1 and T^2. θ represents the probability distribution of the topics T^1 and T^2 per sentence (document) S^1 to S^5. As shown in Figure 4 the ETM algorithm first calculates an enrichment weight (γ) for each sentence in terms of its length. Subsequently, the C matrix is calculated using the M matrix, γ and the clustering outputs: θ and β. The ETM algorithm creates a smoothed dataset that balances initial observations with patterns extracted by the LDA algorithm.

[3]World Heart Organization, International Classification of Diseases: http://www.who.int

Data:
S^1: Patient complained of acute chest pain.
S^2: Patient feels absolutely fine.
S^3: She has hypertension.
S^4: Patient feels fine.
S^5: Patient has hypertension and chest tightness.

1) Data representation:
M matrix:

	absolutely	acute	and	chest	complained	feels	fine	has	hypertension	of	pain	patient	she	tightness
S^1	0	0.449	0	0.362	0.449	0	0	0	0	0.449	0.449	0.253	0	0
S^2	0.618	0	0	0	0	0.499	0.499	0	0	0	0	0.348	0	0
S^3	0	0	0	0	0	0	0	0.532	0.532	0	0	0	0.659	0
S^4	0	0	0	0	0	0.634	0.634	0	0	0	0	0.443	0	0
S^5	0	0	0.484	0.390	0	0	0	0.390	0.390	0	0	0.273	0	0.484

2) LDA clustering

Probability distribution of words per topic (β):

	absolutely	acute	and	chest	complained	feels	fine	has	hypertension	of	pain	patient	she	tightness
T^1	0.001	0.111	0.111	0.001	0.001	0.001	0.001	0.220	0.220	0.111	0.001	0.001	0.111	0.111
T^2	0.077	0	0	0.153	0.077	0.153	0.153	0	0	0	0.077	0.305	0	0

Probability distribution of topics per document (θ):

	T^1	T^2
S^1	**0.339**	0.661
S^2	0.024	**0.976**
S^3	**0.969**	0.031
S^4	0.031	**0.969**
S^5	**0.661**	0.339

Enrichment weight:

	γ
S^1	0.73
S^2	1.1
S^3	1.46
S^4	1.46
S^5	0.73

3) ETM: topic-based smoothing

C matrix:

	absolutely	acute	and	chest	complained	feels	fine	has	hypertension	of	pain	patient	she	tightness
S^1	0.038	0.477	0.028	0.436	0.486	0.074	0.074	0.055	0.055	0.476	0.486	0.401	0.028	0.028
S^2	0.700	0.004	0.004	0.164	0.083	0.663	0.663	0.007	0.007	0.004	0.083	0.676	0.004	0.004
S^3	0.005	0.157	0.157	0.009	0.005	0.009	0.009	0.844	0.844	0.157	0.005	0.016	0.816	0.157
S^4	0.109	0.006	0.006	0.217	0.109	0.851	0.851	0.011	0.011	0.006	0.109	0.876	0.006	0.006
S^5	0.020	0.054	0.538	0.429	0.020	0.039	0.039	0.497	0.497	0.054	0.020	0.349	0.054	0.538

Figure 4: An example for the ETM enrichment representation showing the first three steps of the short text classification model.

5.3. Classification

We used an SVM and a multi-layer neural network (NN) as classification algorithms. In the definition of a learning classifier, the training data were the set of documents D and the classes were medical history versus no medical history. The objective of the SVM was to find a hyperplane in a high-dimensional feature space that distinctly classified the input dataset. By internally employing a kernel trick, it selected the discriminative hyperplane based on the computed support vectors. We used the SVM algorithm with the default parameter settings in scikit-learn[4]. The NN classifier in our experiments used a feed-forward architecture and learned to map the input data to the output labels through a series of nonlinear compositions. For short text classification, the ReLU activation function along with the ADAM solver with two hidden layers of 100 units were used. Compared with other non-linearities, the ReLU activation function learns more quickly in deep architectures with many hidden layers. For the learning of the classification algorithms, we chose 80% of the dataset as the training set and used the remaining for testing.

[4]https://scikit-learn.org/stable/modules/svm.html

5.4. Evaluation measures

To compare the performance of the classifiers, accuracy, precision, recall, and the F1 score were used as the evaluation measures. Precision and recall are useful measures when classes are imbalanced. Precision is a measure of the relevance of the result while recall shows how many truly relevant results were returned. The F1 score is the harmonic mean of precision and recall. These evaluation measures were computed as follows:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{total number of documents}} \tag{7}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{8}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{9}$$

$$\text{F1 score} = \frac{2\ .\ \text{Precision}\ .\ \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

5.5. Experiments

5.5.1. Classification performance

We compared the enrichment in representation obtained by the ETM algorithm with the original representation of short texts (denoted by "Raw") using different numbers of topic clusters. Five, 10, 20, and 50 topic clusters were used. For the ETM, we set n_topics as the number of topics, $\alpha = \frac{50}{n\_topics}$, $\beta = 0.01$, and the number of iterations = 1000. Figure 5 illustrates the accuracy of the ETM approach on clinical short text classification. The best accuracy value for the representation of Raw was 87.10% using the NN classifier. The ETM outperformed the other methods on the representation when it used more than 10 topic clusters. Using SVM with 50 topic clusters slightly improved the representation of Raw with an accuracy of 87.27%. The highest difference between the representation of Raw and that of the ETM method occurred when the NN classifier was used with 10 topic clusters. This difference was approximately 2.3%.

As is shown in Figure 5, with the same number of topic clusters, NN moderately improved the SVM classifier. The highest accuracy using the NN classifier was 89.40% for $n\_topics = 10$, and the highest accuracy using the SVM classifier was 87.72% for the same number of topic clusters. Increasing the number of clusters to 50 did not improve classification performance. Using 10 to 20 clusters yielded the best results on our dataset in terms of accuracy.

Table 2 shows the results in terms of macro-average precision, recall and F1 score to compare the performance of the SVM and NN algorithms on clinical short text classification.

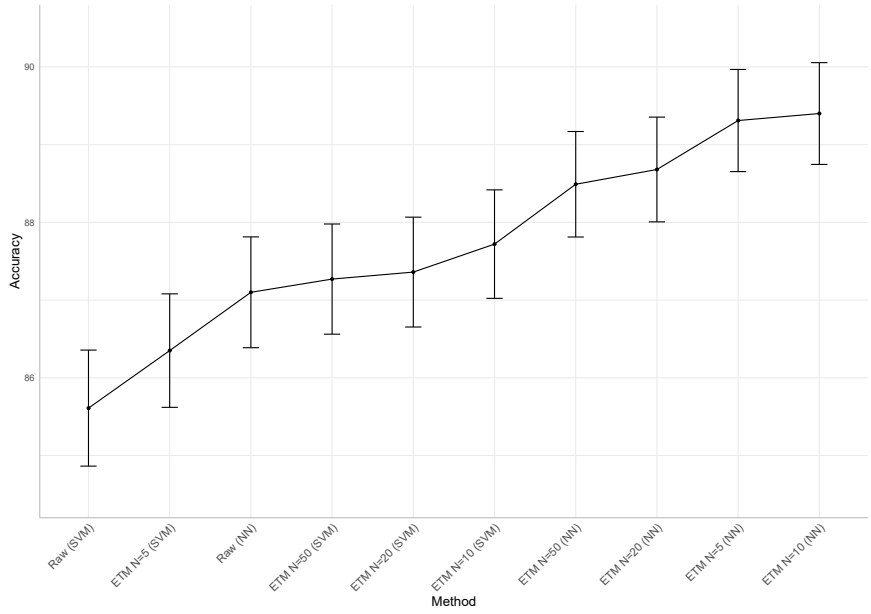

Figure 5: Accuracy results of classification performance of Raw methods and the ETM algorithm using N clusters.

Table 2 shows that the ETM approach improved classification performance considerably compared with Raw in terms of precision and recall. By comparing the precision results, we see that results for Raw were better than those of the ETM using the SVM classifier but inferior to those of the ETM using the NN algorithm.

Table 2 shows that using 10 clusters in the ETM approach yielded the best performance in terms of recall. When $n\_topics = 10$, the SVM and NN classifiers attained recall values of 89.82% and 89.72%, respectively. The NN classifier yielded the best value of 85.79% using the ETM approach with $n\_topics = 5$, and the SVM algorithm obtained a highest value of 83.77% using the ETM approach with $n\_topics = 20$. For nearly all settings, the results remained fairly stable when the number of topic clusters was increased from five to 10, but a slight decline in classification performance was noted when the number of topic clusters was increased from 20 to 50.

These results show that the ETM approach improved classification performance considerably compared with the Raw representation on almost all parameter settings. The ETM approach is robust against changes in the number of topic clusters from five to 20. Even when N was five, the ETM improved the classification performance. This shows its power in enriching representation by using topic clusters.

5.5.2. Evaluation using unlabeled data

The previous sets of experiments employed only labeled sets of UMCU data. By using unlabeled data, the ETM can check whether there are words absent from the labeled set. Tables 3 and 4 show the results of

Table 2: Macro-average precision, recall and F1 score of the assessment of classification performance of Raw methods and the ETM algorithm using N clusters.

Method	Precision	Recall	F1 score
Raw (SVM)	82.21	88.16	85.08
Raw (NN)	82.56	88.47	85.41
ETM $N = 5$ (SVM)	83.00	89.54	86.15
ETM $N = 5$ (NN)	85.79	89.68	87.69
ETM $N = 10$ (SVM)	83.65	89.82	86.63
ETM $N = 10$ (NN)	85.14	89.72	87.37
ETM $N = 20$ (SVM)	83.77	89.57	86.57
ETM $N = 20$ (NN)	84.39	89.06	86.66
ETM $N = 50$ (SVM)	83.02	89.46	86.12
ETM $N = 50$ (NN)	85.01	88.44	86.69

Table 3: Precision, recall, and F1 score of the ETM algorithm using the unlabeled dataset in addition to the labeled set.

Classifier	Class	Precision	Recall	F1 score
SVM	Medical history	77.12	93.62	84.57
	No medical history	88.43	89.36	88.89
NN	Medical history	82.64	94.09	87.99
	No medical history	89.17	91.32	90.23

applying the ETM approach using 10 topic clusters on the unlabeled dataset in addition to the labeled set.

Table 3 shows the results for precision, recall, and the F1 score on the test set for the classes in the dataset. The number of short texts with the label *Medical history* in the test set was 576, and was 1635 for the class label *No medical history*. It is notable that the recall values for the label *Medical history* were the highest for both the SVM and the NN classifiers, where the precision values for the label *No medical history* were significantly higher than the precision for the *Medical history* class. This might have occurred because the number of texts in the first class label was smaller than in the second class label, and thus the percentage of retrieved relevant short texts was lower than the total number of retrieved texts.

Comparing the results in Table 4 with those in Table 2 shows the improvement in the performance of the proposed approach obtained by adding the unlabeled set to the labeled set. It is remarkable that the results for recall were higher than those for precision in both classes and for both classifiers. A higher recall than

Table 4: Micro- and macro-average precision, recall and F1 score of the ETM algorithm using the unlabeled dataset in addition to the labeled set.

Classifier	Metric	Precision	Recall	F1 score
SVM	Micro-avg.	81.97	83.66	82.81
	Macro-avg.	82.78	91.49	86.73
NN	Micro-avg.	86.25	86.47	86.36
	Macro-avg.	85.91	92.71	89.11

precision means that the classifier tends to extract more of the relevant outputs rather than retrieving correct outputs. As shown in Table 2, the highest values for precision and recall without the unlabeled dataset were 85.79% and 89.82%, respectively. The former was obtained for the ETM with five clusters using the NN classifier and the latter is for the ETM with 10 clusters using the SVM classifier. The highest macro-average precision and recall of the ETM approach in this experiment were 85.91% and 92.71%, respectively, obtained using the NN classifier with 10 topic clusters. Table 4 shows that the NN classifier obtained better results than the SVM classifier. The highest F1 score for the ETM using NN classifier was 90.23% and that for it with the SVM classifier was 88.89%. Both the NN and SVM classifiers were influenced by the enrichment of the dataset through the unlabeled data. Thus, when more data are available, the models have a higher chance of improving performance. Moreover, with more data for clinical text classification, the chances of encountering new words in new samples decrease.

5.5.3. Comparison study: Crest, CNN, and ETM

We compared the results of the ETM algorithm with the following two methods:

Crest: Crest [33] generates topic clusters from training data by exploiting a clustering method, and then uses the topic information to extend the representation of short texts. This approach is similar to that of the ETM as it uses a clustering method. The difference is that Crest uses the cosine similarity between a short text and a topic cluster as similarity vector that is then used for text representation. The ETM does not use a similarity metric but the probability distributions of documents and topics inferred from the generative process of the LDA. Crest increase the dimensions of the feature space by the number of clusters whereas the ETM uses the same dimensions as the training set.

CNN: As mentioned in Section 2, Hughes et al. [25] implemented a deep CNN model for medical text classification at the sentence level. A CNN model requires that the length of the text have a fixed size as input. Therefore, they chose a maximum word length of 50 for a text, and applied a Word2vec layer of size 100. Their model consisted of two sets of convolutional layers followed by two max pooling layers. They used convolutional filters and applied a dropout function to help prevent overfitting. Then, a fully connected layer with 128 units was followed by a dense layer using a softmax function.

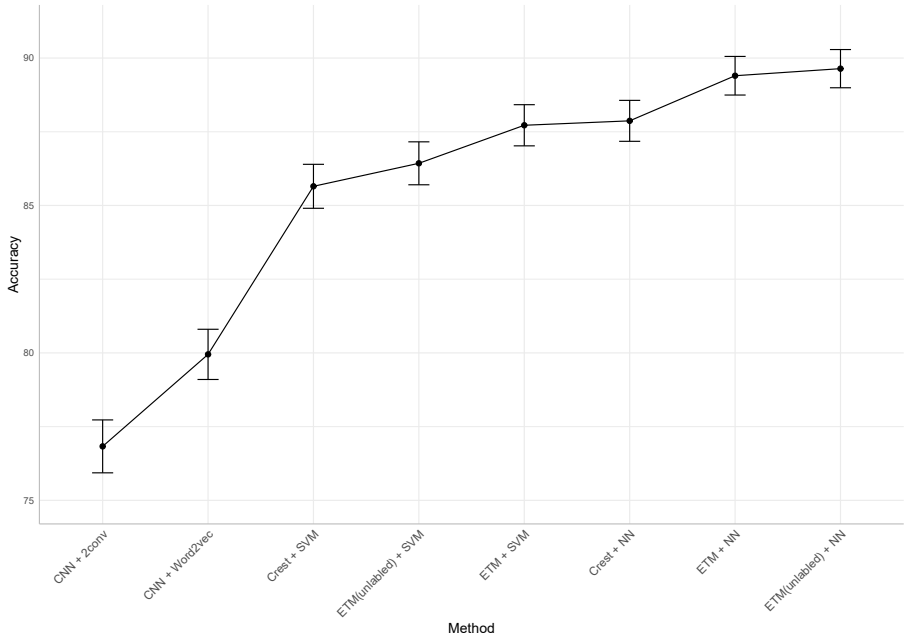

Figure 6: Comparison study: Crest, CNN, and ETM

Figure 6 shows the results of a comparison among Crest, CNN, and ETM. In the experiments, the CNN model was used with two settings: one experiment used two convolutional layers (*'CNN + 2conv'*) and the other, *'CNN + Word2vec'*, applied the CNN approach using two pairs of convolutional layers followed by two max-pooling layers and dense layers. As shown in the figure, the CNN models had lower accuracy than the Crest and ETM. There are two reasons for this: (1) Feature engineering in the Crest and ETM approaches has been proposed especially for the short text classification problem. (2) The trained word vectors are not rich enough to capture the semantics and diversity in our clinical text collection of Dutch language. While there is no publicly available pre-trained word vectors for Ducth clinical text, Dutch word vectors trained on social media and Wikipedia can be experimented as initial weights for the deep learning models in future work.

The highest accuracy in the experiments on the CNN models using the test set was 79.95% whereas the closest model to this was that of Crest using the SVM classifier, with a value of 86.65%. The models with enriched representations delivered better performance than the CNN classifier, which proves the effectiveness of using smoothing methods to enrich the original representation. The accuracy of Crest using the NN classifier was higher than that of the ETM using the SVM algorithm. The differences between *'Crest + NN'*, and *'ETM(unlabeled) + SVM'* and *'ETM + SVM'* were 1.44% and 0.15%, respectively. This shows the positive effect of using a neural network approach compared with an SVM classifier. In all approaches used in these experiments, the NN classifier had an accuracy of approximately 2% higher than the SVM classifier. The highest accuracy was obtained by the ETM method, 89.64%, when it used the unlabeled dataset with the NN classifier. This shows the power of the ETM approach in overcoming the brevity and sparsity of

short texts by utilizing topic clusters extracted from the training data for better representation.

6. Conclusions

EHRs usually store patients' disease history in free text form. Although this lack of structure might not directly affect patient care in clinical settings, it does affect other uses of the EHR, such as patient recruitment for clinical trials. Automated text analysis using text mining algorithms eliminates administrative burdens and is important for research. The textual classification of clinical sentences is a first step in the automated extraction of medical history. Because of the limited number of words used in clinical sentences, this problem is considered that of short text classification. Current approaches to clinical short text classification mainly use external dictionaries but this has a number of drawbacks, including the lack of a universal medical dictionary for different languages. This study proposed an unsupervised model-based smoothing method, the ETM approach, that uses an internal knowledge acquisition mechanism without employing any external dictionary. The ETM considers the length of each document in the enrichment phase and adds hidden information behind the topic clusters gained from the clustering algorithm. It is notable that the purpose of the enrichment is to *improve the text classification workflow*; we do not change the original record or the results displayed to a physician. While model interpretability is difficult to achieve in practice, using BOW representation with the ETM approach makes prediction explainable. To mitigate the error in short text classification, we trained the enriched representation on the SVM and NN classification algorithms, and used clinical cardiovascular notes from the UMCU hospital in the Netherlands. Experimental results showed that applying the proposed ETM approach delivers good classification performance, and is comparable to prevalent alternatives. Moreover, it is simple and easy to implement, where this makes the ETM a promising tool for the analysis of short texts for various applications. In future work, we plan to look into the performance of the ETM approach in prognostic prediction models by incorporating other variables from EHRs. Furthermore, we will study the impact of the size of the dataset on performance and investigate the use of enriched representations in complex deep learning models.

References

[1] D. Demner-Fushman, W. Chapman, C. McDonald, What can natural language processing do for clinical decision support?, Journal of Biomedical Informatics 42 (5) (2009) 760–772.

[2] M. Sevenster, J. Bozeman, A. Cowhy, W. Trost, A natural language processing pipeline for pairing measurements uniquely across free-text ct reports, Journal of Biomedical Informatics 53 (2015) 36–48.

[3] J. Jonnagaddala, S. Liaw, P. Ray, M. Kumar, N. Chang, H. Dai, Coronary artery disease risk assessment from unstructured electronic health records using text mining, Journal of Biomedical Informatics 58 (2015) S203–S210.

[4] M. Ghassemi, T. Naumann, F. Doshi-Velez, N. Brimmer, R. Joshi, A. Rumshisky, P. Szolovits, Unfolding physiological state: Mortality modelling in intensive care units, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 75–84.

[5] C. Friedman, L. Shagina, Y. Lussier, G. Hripcsak, Automated encoding of clinical documents based on natural language processing, Journal of the American Medical Informatics Association 11 (5) (2004) 392–402.

[6] R. Byrd, S. Steinhubl, J. Sun, S. Ebadollahi, W. Stewart, Automatic identification of heart failure diagnostic criteria, using text analysis of clinical notes from electronic health records, International Journal of Medical Informatics 83 (12) (2014) 983–992.

[7] M. Torii, J. Fan, W. Yang, T. Lee, M. Wiley, D. Zisook, Y. Huang, Risk factor detection for heart disease by applying text analytics in electronic medical records, Journal of Biomedical Informatics 58 (2015) S164–S170.

[8] A. Khalifa, S. Meystre, Adapting existing natural language processing resources for cardiovascular risk factors identification in clinical notes, Journal of Biomedical Informatics 58 (2015) S128–S132.

[9] M. Kozlowski, H. Rybinski, Clustering of semantically enriched short texts, Journal of Intelligent Information Systems 53 (1) (2019) 69–92.

[10] Y. Shen, Q. Zhang, J. Zhang, J. Huang, Y. Lu, K. Lei, Improving medical short text classification with semantic expansion using word-cluster embedding, in: International Conference on Information Science and Applications, Springer, 2018, pp. 401–411.

[11] S. Zelikovitz, H. Hirsh, Improving short text classification using unlabeled background knowledge to assess document similarity, in: Proceedings of the seventeenth international conference on machine learning, 2000, pp. 1183–1190.

[12] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, M. Demirbas, Short text classification in twitter to improve information filtering, in: ACM 841–842. Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, 2010, pp. 841–842.

[13] X. Cheng, X. Yan, Y. Lan, J. Guo, Btm: Topic modeling over short texts, IEEE Transactions on Knowledge and Data Engineering 26 (12) (2014) 2928–2941.

[14] C. Yin, L. Shi, J. Wang, Short text classification technology based on knn + hierarchy svm, in: Springer, Advanced Multimedia and Ubiquitous Engineering, May 22–24, 2017, pp. 633–639.

[15] M. M. Mirończuk, J. Protasiewicz, A recent overview of the state-of-the-art elements of text classification, Expert Systems with Applications 106 (2018) 36–54.

[16] P. Unnikrishnan, V. Govindan, S. M. Kumar, Enhanced sparse representation classifier for text classification, Expert Systems with Applications 129 (2019) 260–272.

[17] S. Cao, B. Qian, C. Yin, X. Li, J. Wei, Q. Zheng, I. Davidson, Knowledge guided short-text classification for healthcare applications, in: Proceedings of IEEE International Conference on Data Mining (ICDM), 2017, pp. 31–40.

[18] D. Blei, A. Ng, M. Jordan, Latent dirichlet allocation, Journal of machine Learning Research 3 (1) (2003) 993–1022.

[19] G. Mujtaba, L. Shuib, N. Idris, W. L. Hoo, R. G. Raj, K. Khowaja, K. Shaikh, H. F. Nweke, Clinical text classification research trends: systematic literature review and open issues, Expert Systems with Applications 116 (2019) 494–520.

[20] L. Yao, C. Mao, Y. Luo, Clinical text classification with rule-based features and knowledge-guided convolutional neural networks, BMC Medical Informatics and Decision Making 19 (3) (2019) 71.

[21] S. Kocbek, L. Cavedon, D. Martinez, C. Bain, C. Mac Manus, G. Haffari, I. Zukerman, K. Verspoor, Text mining electronic hospital records to automatically classify admissions against disease: measuring the impact of linking data sources, Journal of Biomedical Informatics 64 (2016) 158–167.

[22] D. D. A. Bui, Q. Zeng-Treitler, Learning regular expressions for clinical text classification, Journal of the American Medical Informatics Association 21 (5) (2014) 850–857.

[23] S. J. Fodeh, D. Finch, L. Bouayad, S. L. Luther, H. Ling, R. D. Kerns, C. Brandt, Classifying clinical notes with pain assessment using machine learning, Medical & biological engineering & computing 56 (7) (2018) 1285–1292.

[24] A. Blanco, A. Casillas, A. Pérez, A. D. de Ilarraza, Multi-label clinical document classification: Impact of label-density, Expert Systems with Applications 138 (2019) 112835.

[25] M. Hughes, I. Li, S. Kotoulas, T. Suzumura, Medical text classification using convolutional neural networks, Stud Health Technol Inform 235 (2017) 246–250.

[26] Y. Lv, Y. Deng, M. Liu, Y. Cui, Q. Lu, Short text classification of emr based on entities and dependency parser, Zhongguo yi liao qi xie za zhi= Chinese journal of medical instrumentation 40 (4) (2016) 245–249.

[27] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882.

[28] D. Jurafsky, J. H. Martin, Speech and Language Processing: An introduction to speech recognition, computational linguistics and natural language processing, 3rd Edition, Prentice Hall, 2019.

[29] C. C. Aggarwal, Machine learning for text, Springer, 2018.

[30] D. Bollegala, V. Atanasov, T. Maehara, K. Kawarabayashi, Classinet–predicting missing features for short-text classification, arXiv preprint arXiv:1804.05260.

[31] M. Chen, X. Jin, D. Shen, Short text classification improved by learning multi-granularity topics, in: AAAI, Twenty-Second International Joint Conference on Artificial Intelligence, 2011, pp. 1776–1781.

[32] S. Yang, W. Lu, D. Yang, L. Yao, B. Wei, Short text understanding by leveraging knowledge into topic model, in: Association for Computational Linguistics. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, 2015, pp. 1232–1237.

[33] Z. Dai, A. Sun, X. Liu, Crest: Cluster-based representation enrichment for short text classification, in: Springer, Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2013, pp. 256–267.

[34] M. Kozlowski, H. Rybinski, Semantic enriched short text clustering, in: International symposium on methodologies for intelligent systems, Springer, 2017, pp. 435–445.

[35] S. Yang, G. Huang, B. Cai, Discovering topic representative terms for short text clustering, IEEE Access 7 (2019) 92037–92047.

[36] J. Y. Lee, F. Dernoncourt, Sequential short-text classification with recurrent and convolutional neural networks, arXiv preprint arXiv:1603.03827.

[37] F. Hill, K. Cho, A. Korhonen, Learning distributed representations of sentences from unlabelled data, arXiv preprint arXiv:1602.03483.

[38] W. W. Cohen, Integration of heterogeneous databases without common domains using queries based on textual similarity, in: ACM SIGMOD Record, Vol. 27, ACM, 1998, pp. 201–212.

[39] V. Menger, F. Scheepers, L. M. van Wijk, M. Spruit, Deduce: A pattern matching method for automatic de-identification of dutch medical text, Telematics and Informatics 35 (4) (2018) 727–736.

[40] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, " O'Reilly Media, Inc.", 2009.

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