USING DEEP CONVOLUTIONAL NEURAL NETWORKS WITH SELF-TAUGHT WORD EMBEDDINGS TO PERFORM CLINICAL CODING

Arnon Bruno Ventrilho dos Santos, Yohan Bonescki Gumiel, Deborah Ribeiro Carvalho
Pontifícia Universidade Católica do Paraná (PUC-PR)
E-mails: asantos.quantum@gmail.com, yohan.gumiel@gmail.com, drdrcarvalho@gmail.com

Abstract: Clinical coding represents the transposition of clinical findings and diagnostics into codes contained in the International Classification of Diseases (ICD). This represents a very important task for the standardization of disease diagnoses and payment of clinical bills. To perform such task, hospitals assign the role of "clinical coder" to the person responsible for reading the whole clinical documentation and assigning the ICD codes accordingly. This task, however, is very time-consuming and the uncertainty that is related to natural language can introduce mistakes in coding. It is also known that wrong coding can lead to delays in paying process, and in some cases financial and legal disruption. The objective of this research is to propose a model to automate Clinical Coding by using clinical discharge summaries. These texts, written in Brazilian Portuguese, were transformed into word embeddings and then fed into a classifier based on a Deep Convolutional Neural Net. Given the imbalance in data, we've trained and tested the model using a stratified *k-fold* approach (k = 10) with cost-sensitive learning, obtaining on our best model an average *F-score* of 0.97 with standard deviation of 0.04. We also tested the model against a balanced augmented database, obtaining 82,9% of final accuracy. These results show that our model outperforms some of the recent models developed for similar tasks. Since we have not taught the algorithm any rules of language or coding, these results suggest that clinical coding can be automated by Deep Learning based approaches that uses self-taught word embeddings.

Keywords: Learning systems, supervised learning, unsupervised learning, machine learning, natural language processing.

1. INTRODUCTION

Clinical coding is the process where part of the clinical information of a patient, usually stored in clinical narratives, is transformed into codes (Preda, Chiriac and Musat, 2012). This process is also described as the translation and grouping of clinical concepts into codes that aim to standardize the nomenclature of symptoms, diagnoses and other clinical situations in a single language (Aalseth, 2006). Among all the coding standards available, the official one and widely used is the International Classification of Diseases (ICD) codes (WHO, 2017; Laurenti, 1991). The data used for coding are texts written by clinicians in natural language. These texts contain the patient information and general details about its stay at the healthcare institution, as well as clinicians' recommendations and the situation in which the patient was released from the hospital. Usually, these narratives are coded through a manual process. This process involves reviewing the clinical documentation of the patient by a human agent, the "clinical coder": This person, usually a physician or other clinical professional, is responsible for applying the ICD codes according to what is described in the clinical documentation, performing the process known as clinical coding (Stanfill et al., 2010). It is important to emphasize the importance of clinical coding as one of the main components of the coordination process of all the actors of the health system who are involved in the provision or financing of health services. This task has a great impact on the financial activities of health care providers, in monitoring their activities and on the evaluation and estimation of the need for health services (Preda, Chiriac and Musat, 2012). It is also an important aid tool in epidemiological research, since through the standardization of clinical information that the

coding enables, it is possible to identify how public health situations are distributed among a population (Aalseth, 2006).

The efforts to use natural language processing (NLP) for automating clinical text processing have resulted in important advances in automating clinical coding. Some private companies have developed tools (known by the acronym "CAC", computer automated coding) that, using NLP, extract information from clinical texts and encode them, providing some automation to for clinical coding. The task of understanding texts through PLN, however, is a traditionally difficult problem due to the extreme variability of language formation, differentiation between languages and meanings of words (Zhang and LeCun, 2016; Pacheco, Nohama and Schulz, 2013). In addition, traditional PLN methods require that the features extracted from the texts be manually defined and adjusted according to the nature of the problem [8], making this a completely empirical process and dependent of the researcher expertise on the context of the problem (Yang et al., 2013; Collobert and Wetson, 2008). Particularities such as these make PLN, in a sense, specialized in the language it was developed to, so that if language is changed, many features need to be redesigned manually (Zhang and LeCun, 2016).

An alternative to the traditional PLN process is the use of numerical representations of words, known as word-embeddings, which use the distributive hypothesis from linguistic to capture the semantic context in which words are inserted. These numerical representations, usually extensive and non-sparse, are organized into vectors commonly used as input into machine learning classifiers (Pennington, Socher and Manning, 2014). Advancements in the field of machine learning and in the computational capacity enabled the appearing of new strategies of extracting information from unstructured data, such as in texts and images. The main exponent of these advancements is the technique called Deep Learning (DL) (Santos and Carvalho, 2015). DL is often referred to as a subfield of machine learning that makes use of artificial neural networks of multiple layers and that deals with the recognition, processing, interpretation and classification of images, texts, speech, etc. by learning through representation (Santos and Carvalho, 2015).

DL methods deals with multi-level representation learning, obtained through the composition of simple but non-linear modules that transform simple representations (at higher levels) into increasingly complex representations, insofar as the representation levels deepen. With the composition of such transformations, very complex functions can be learned (LeCun, Bengio and Hinton, 2015).

Significant advancements in sentiment analysis in social networks and even in the understanding language without previous knowledge about its characteristics were obtained from the use of DL and its effectiveness demonstrated in comparison to other techniques of machine learning (Santos and Gatti, 2014). Several studies that use DL for NLP have appeared over the years in order to overcome some of the difficulties encountered in the manual process of defining characteristics and optimizing the results initially found with NLP (Zhang and LeCun, 2016; Hermann et al., 2015). Kim (2014) demonstrated a method for sentence classification using DL with word embeddings, which is considered by Rios and Kavuluru (2015), Lenc and Král (2017) and by Zhang and Wallace (2016) as the baseline method for sentence and document classification.

Thus, DL for NLP with word embeddings, besides being a modern approach to a traditionally complex problem, represents a potential solution to the problem of clinical coding.

2. CLINICAL CODING

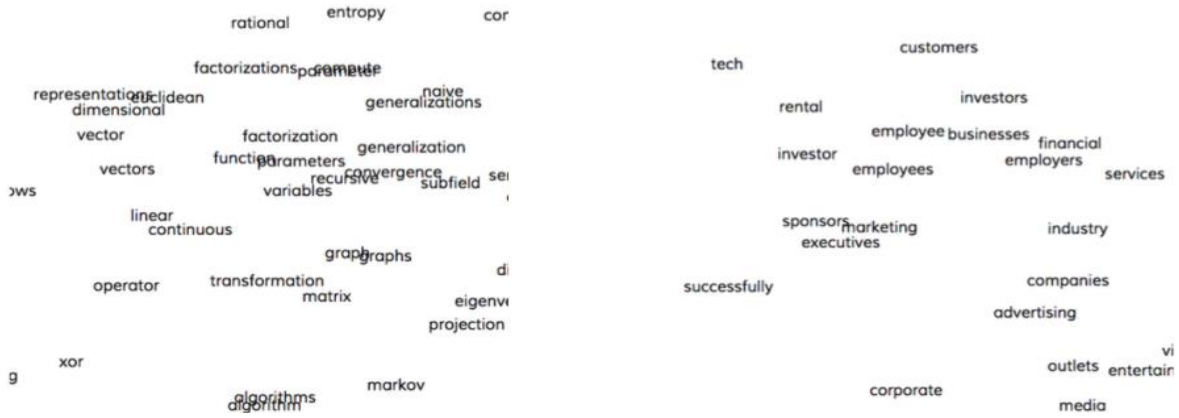
Clinical coding refers to the task of reading the clinical narratives, identify concepts such as the main diagnosis, additional diagnoses of comorbidity or complication, basic causes, surgical procedures and associate each of these concepts into an ICD code (Lopes, 2009). Unstructured data in text format constitute the clinical narratives (Chu, 2002). These data are

part of the evolution record or clinical history of the patient, which, when filled in correctly, alerts on variations and results of the patient's consultations, diagnoses, medications and behavior (Pacheco, Nohama and Schulz, 2013).
The clinical coder, therefore, is the professional assigned to read the information contained in the clinical narratives. If the clinical record is incomplete or inaccurate, the coder may find difficulties in assigning the ICD code, which potentially leads to error in coding. It is also important that the coding process be as accurate as possible to avoid the health institution to be investigated under fraud accusation or even be affected by financial penalties (Lopes, 2009).
Among the documents available for coding on those clinical narratives, there is the discharge summary. This document condenses medical information of the patient and facilitates its eventual readmission or consultation in the hospital. It contains, at least, signs and symptoms of the patient, personal and family history, physical examination, reports, medications used and plans for the follow-up of the case. Strategies for extracting information in clinical narratives, such as NLP, can make use of this document (Pacheco, Nohama and Schulz, 2013). There is several information in the discharge summary, ranging from the attendance to the procedures performed that can benefit patients, as professionals use this information for decision making within a health system, avoiding duplication of exams or procedures and may decrease the cost of care (Pacheco, Nohama and Schulz, 2013).
There are researches that demonstrate the usage of traditional NLP for extracting information from clinical texts such as the discharge summary (Pacheco, Nohama and Schulz, 2013). However, authors such as Pennington, Socher and Manning (2014) criticizes the effectiveness of conventional NLP strategies, claiming that they have two common shortcomings for natural language processing in any domain, namely the simplification of language assumptions and the need for hand designed features, which leads traditional NLP strategies to be adapted to the language it was designed for.

3. WORD EMBEDDINGS

According to Pennington, Socher and Manning (2014), models that make use of word-embeddings represent an important alternative to the traditional NLP strategies. A word embedding is a mapping from words to vectors of real numbers, capturing the semantic relationship between words based on the similarity of those vectors (Jiang et al., 2018). One advantage of using this method for NLP is that it is not dependent on the language and a certain vocabulary, being able to capture the semantic meaning of words within a space based on the distributional hypothesis of Harris (Harris, 1954). According to this hypothesis, if we observe two words that constantly occur within the same contexts, it is possible to assume that they mean similar things. Note that the hypothesis does not require words to occur together, it only requires that words occur within the same set of other words. Let's take the words "swim" and "swimming" as examples. According to Harris (1954), these two words must carry similar meaning because they often occur with the same neighboring words. The viability of the distributive hypothesis has been demonstrated in numerous experiments (Rubenstein and Goodenough, 1965). The general idea behind word-embeddings models is motivated by this distributive hypothesis, producing vector spaces with several dimensions, in which words are represented by context vectors whose relative orientations are assumed as indicators of semantic similarity (Heuer, 2016). The objective of this distributive hypothesis within NLP is to find a representation vector that approaches the meaning of a given word, thus avoiding the traditional NLP process.

Figure 1: Representation of words in a two-dimensional space. The proximity of words represents the semantic similarity between them given the context in which word are inserted



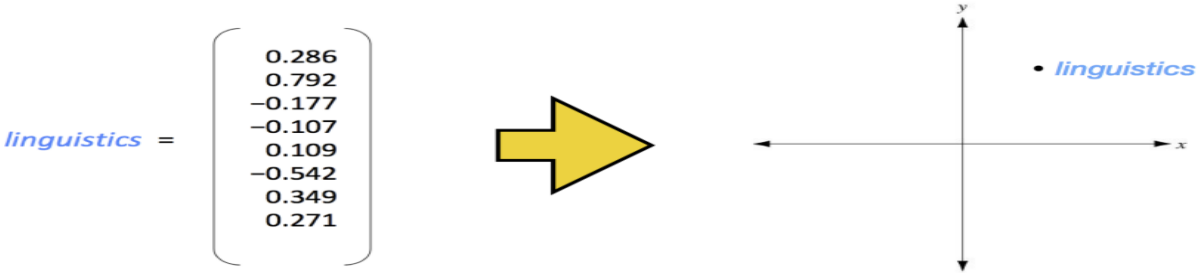
Source: Heuer (2016)

To create these representations, some algorithms are available. Pennington, Socher and Manning (2014), used 50 billion words extracted from texts from multiple sources to create a semantic relationship algorithm known as "GloVe" (acronym for Global Word Vectors). Mikolov et al. (2013) used about 6 billion to design the word2vec algorithm. Pennington, Socher and Manning (2014) even compared GloVe to Word2Vec, concluding that the GloVe algorithm can produce the best word embeddings, getting greater accuracies and faster. The same results were verified by Pereira et al. (2016), Rodríguez (2016) and by Dhingra et al. (2017), while, Berardi, Esuli and Marcheggianni (2015) and also Kang et al. (2016) point to better accuracy using Word2Vec algorithm.

What both experiences demonstrated is that, given enough words in a vocabulary, the semantic relationship between them can be extracted by applying an algorithm to create a word-embedding. The semantic relationship can then be visualized after applying dimensionality reduction strategies and plotting the words in respect to their cosine similarity. Although this provides a nice visualization of the word embeddings (Figure 1), one needs to notice that dimensionality reduction could affect the meaning of words.

Word-embeddings model also offers another important feature, which makes it particularly interesting for machine learning classification tasks. The fact that word vectors are represented by numbers (Figure 2) makes this model a qualified input for machine learning models that receive as input numerical data, such as artificial neural networks (Pennington, Socher and Manning, 2014; Heuer, 2016).

Figure 2: Word-embeddings representing the word “linguistics”



4. DEEP LEARNING

Deep Learning (DL) is a subcategory of machine learning which deals with neural networks with multiple layers of processing to learn data representations with multiple levels of abstraction, which is currently the state-of-art method in computer vision and nature language processing (LeCun, Bengio and Hinton, 2015; Sakhavi, Guan and Yan, 2018; Goodfellow, Bengio and Courville, 2016). Its origin refers to the first multi-layer perceptron (MLP) by Ivakhnenko, whose purpose was the solution of problems that grew in complexity and level of abstraction (Santos and Carvalho, 2015). DL, has also been extensively studied and applied in the identification of complex patterns such as face recognition in social networks, word processing, manuscripts, oral communication, human-computer interaction, and health with results that sometimes outperform human agents in tasks that demand the identification of patterns (Santos and Carvalho, 2015).

The main feature of DL and what makes this area so relevant to machine learning is its capacity to deal with large amounts of data as input and its ability to learn through representations. Representation learning allows an algorithm to receive raw data as input and automatically discovers the representations required to detect or classify patterns as representations are transferred between layers (LeCun, Bengio and Hinton, 2015). This feature of DL models is crucial for image recognition and NLP tasks, since the features contained in this kind of data are very complex to design by hand.

The literature demonstrates that DL is constantly being used for NLP tasks. Lv et al. (2016) used an auto-encoder based Deep Learning approach to capture relation in clinical texts written in natural language. Ayyar and Walk (2016) used a recurrent neural network (RNN) to tag patient notes with ICD-9 codes, achieving F-score 0.708. Zhang and LeCun (2016) used convolutional neural nets (CNN), which represent successive computational layers alternating between convolution and subsampling (Pang et al., 2018), to learn language from scratch using a massive database of texts. Hughes et al. (2017) used CNN for medical documents classification using pre-trained word-embeddings achieving 0.68 accuracy on classification. Rios and Kavuluru (2015) compared their one-layered CNNs with 12 other methods to classify clinical documents, demonstrating that this approach outperforms others such as support vector machines, naïve bayes and logistic regression by achieving F-score 0.721. Lenc and Král (2017) used CNN with word embeddings for multi-label document classification, achieving F-score of 0.847. While most models use CNNs for sentence and text classification, Rios and Kavuluru (2015), Lenc and Král (2017) and Zhang and Wallace (2016) suggest that Kim (2014) should be used as the baseline models for such tasks.

The advantage of DL using CNNs for NLP is that these models can work with pre-trained word vectors (such as word2vec or GloVe) that can be easily downloaded from the internet or learned from within the dataset (Hughes et al., 2017). Therefore, DL algorithms do not need to learn the language structure or the characteristics of the language, which in the traditional NLP process need to be hand-designed, but only to learn the context in which the words appear to be able to assign a meaning (Ayyar and Walk, 2016; Hughes et al., 2017; Pennington, Socher and Manning, 2014). Hughes et al. (2017) and Pennington, Socher and Manning (2014) state that word vector can optimize the performance of DL models for PLN.

However, for using pre-trained word embeddings downloaded from the internet, one needs to consider the nature of the problem. We see from our database that some clinical-specific words are written the same way words in Portuguese with very different meanings are. For example, the word “ITU”, meaning “urinary tract disease” in Portuguese, is also the name of a city in Brazil. Therefore, a pre-trained word embedding composed of non-clinical texts may lead the model to learn representations that will not match with the context being presented on the embeddings, degenerating the model’s accuracy. This might have happened in Ayyar and Walk (2016).

5. DATA

It is widely known that DL models tend to overfit when using relatively small datasets (Asperti and Mastronado, 2017), however, it is not well understood the "minimal" dataset size for successful DL experiments. Rios and Kavuluru (2015) use approximately 90.000 examples for a CNN based model, Hughes et al. (2017) used approximately 15.000 examples for a similar approach and Mou et al. (2016) even used datasets with approximately 4400 examples for NLP with CNN. All these experiments do not demonstrate an overfitting pattern, which is commonly identified by high accuracies in training set, with low accuracies in test set or even by high loss in test with very low loss in training (Subramanian and Simon, 2013). But even when an experiment demonstrates an overfitting pattern, there are some strategies useful to mitigate this issue. One possibility is to use data augmentation to generate more data for training (Asperti and Mastronado, 2017; Ganganwar, 2012), other common approach is to use regularization strategies such as dropout for DL models (Goodfellow, Bengio and Courville, 2016; Srivastava et al., 2014).Observational and controlled studies were included in this study, and whose results could be confirmed by expert analysis in the health area and computer science, as seen along some of these studies.

For this research, we used 4030 discharge summaries as our dataset. While this can represent a "small dataset", we analyze overfitting patterns and tried to mitigate them with strategies mentioned in the next section. A number of 2030 of those discharge summaries relate to the diagnoses of "Other Urinary Tract Disorders" (OUTD), from 1 hospital in Brazil. These documents are coded with labels according to the codes in Table 1. This database was made available in compressed file, with discharge summaries in .pdf format. In addition, a further 2.000 discharge summaries, in .txt format, was added to the database comprising other diagnoses that are different from the ones mentioned above and labeled with a single code (N39.5) to differentiate these examples from the ones exclusively used for OUTD. Table 2 demonstrates the distribution of the discharge summaries contained in both bases, as well as the code assigned to these documents. From this perspective, it is notable that our dataset is very imbalanced.

Table 1: ICD-10 Codes for "Other Urinary Tract Disorders"

ICD-10 Code	Description
N39.0	Urinary tract infection of unspecified location
N39.1	Unspecified persistent proteinuria
N39.2	Unspecified orthostatic proteinuria
N39.3	"Incontinence of tension ("stress")"
N39.4	Other specified urinary incontinence
N39.8	Other specified urinary tract disorders
N39.9	Unspecified disorders of the urinary tract

Table 2: Samples in the dataset for each class

Class	Number of Samples
N39.0	1762
N39.1	11
N39.2	0
N39.3	47
N39.4	51
N39.5	2000
N39.8	12
N39.9	147

6. METHOD

For this research, we aimed to measure the performance of the baseline model of Kim (2014) shown in Figure 3 for the problem we are addressing and propose any adaptation that could optimize the model. We then expanded the experimentation by using modified CNNs, similar to what was proposed by (Hughes et al., 2017; Rios and Kavuluru, 2015; Lenc and Král, 2017). With our test scenarios with different CNNs, we aim to provide a further experimentation outlook on the differences between CNN models and how they influence NLP multi-class classification metrics, while suggesting a model configuration adapted to automating clinical coding for a given set of ICD codes.

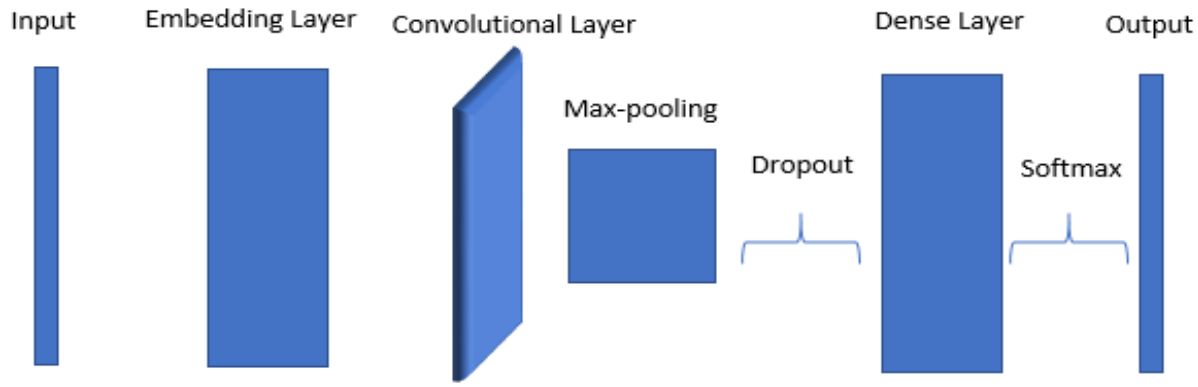
Our model uses DL to classify hospital discharge summaries according to the its specific codes in ICD. For this purpose, we use word-embeddings obtained from the database itself. These embeddings are what we call “self-taught” word embeddings. To create these embeddings, we used the programming language Python to extract the text contained in those 4030 discharge summaries. We then tokenize these texts, removed stopwords in Portuguese and normalized the text. With this data we then used the "glove-python" library. This Python library implements the GloVe algorithm to create word vectors based on co-occurrence of words within a context. The choice of the GloVe algorithm was due to the work of Pennington, Socher and Manning (2014) which, comparing GloVe with other algorithms such as word2vec found that GloVe more efficiently captured the semantic meaning of the words and needed less training time. The choice of developing a “self-taught” word embedding is motivated by the fact that clinical vocabulary has specificities (such as acronyms) that cannot be obtained using pre-trained embeddings from journalistic texts, traditionally used for this matter (Ayyar and Walk, 2016; Li and Huang, 2016). Thus, the self-taught embedding proposed here has the semantic characteristics of the clinician vocabulary in Brazilian Portuguese. To test the influence of word vectors of different sizes (dimensions) on the final classifier, we developed with this research embeddings with 50, 100, 300 and 500 dimensions, as also proposed by Pennington, Socher and Manning (2014). For the development of the DL model, the approximation suggested by Zhang and Wallace (2016) was performed, where the baseline CNN model is implemented (Kim, 2014) and, from the results of preliminary classification, we search for optimal parameters until we obtained the best performance of the model.

Five different CNN models were trained for each of the 4 different self-taught embedding sizes, varying from 1 to 5 convolutional layers with different window sizes, resulting in

twenty different models. For training the model, we used the stratified k-fold strategy (k=10), along with cost-sensitive learning (Ganganwar, 2012), due to the high imbalanced dataset. This stratified version of the *k*-fold algorithm allows the 10 divisions of the dataset to have the same class representation as the original database. This also allows to use the whole database for training and testing, since for each training cycle, 1/10 of the base is used for testing and 9/10 for training. Thus, at the end of the 10 folds the entire dataset was used for training and for testing. This measure, associated cost-sensitive learning (Ganganwar, 2012), aims to ensure that the proposed algorithm has a greater cost penalty for erroneously classified examples on minority classes, so that these errors are less frequent, as suggested by Goodfellow, Bengio and Courville, 2016 and by Huang et al. (2016). From the initial approximations and exploration suggested by Zhang and Wallace (2016), we achieved optimum results with a multilayer CNN model trained for 20 epochs with hyperparameters optimized using the "Adam" algorithm (Kingma and Ba, 2015; Yan, Sakhavi and Guan, 2018).

Figure 3 illustrates the baseline CNN model from which we derive the models we tested. Our best model, however, is composed with multiple CNNs (Fig. 10). It receives the input (discharge summaries with length *L*), convert it into word vectors on the embedding layer (*e*), and send the vectors to the CNN, which has 128 filters (k), and windows (*w*) of different sizes when dealing with multiple CNN layers. This measure is desirable to obtain different patterns from each convolution (Ganganwar, 2012). The convolution layer(s) are activated with "ReLu' function [33], followed by a 1-maxpooling layer, which aims to identify the signals activated with greater force during the convolution (Goodfellow, Bengio and Courville, 2016; Zhang and Wallace, 2016). When multiple CNN layers are applied, we also use a new layer of global max pooling after the last CNN layer. Global max pooling subsamples the signals activated throughout the whole convolution and 1-max-pooling process, feeding a dense classifier with size (*i*) 128 and dropout (*d*) of 0.2, which serve us as a regularization measure to smooth any type of overfitting [35]. The output of this layer is a softmax layer that provides a probability distribution between the 7 possible classes (6 of the ICD and 1 general).

Figure 3: The baseline model



Source: Kim (2014)

In addition to these measures, we used early-stopping (Loughrey and Cunningham, 2005; Ganganwar, 2012) during training, which monitors the evolution of the cost function during the test. If the value of the cost function is not reduced for 5 epochs, the training in that fold is interrupted, the weights are updated the training moves to the next fold.

7. EVALUATION

To evaluate the model, we used an approach similar to that used by Ayyar and Walk (2016) and by Hermann et al. (2015), who used the F-score measure to evaluate the quality of the model. F-score relates precision and recall (specificity and sensitivity) to produce a measure that precisely describes the quality of the texts classification within an unbalanced dataset (Sokolova and Lapalme, 2009). We also use precision and recall to analyze the best model even further (Sokolova and Lapalme, 2009).

Since the discharge summaries used on this research are already coded, that is, there is already a classification performed by a clinical coder and will be tested by the model proposed, the results predicted by the model were compared to the codes already assigned by the coders to evaluate the results of the proposed classifier during the model development stage.

To evaluate the model even further, we used a data augmentation strategy for texts to create an artificial test dataset, based on real examples of the from the dataset used for training, as once proposed by Wang and Yang (2015). This test dataset is totally balanced, evenly distributed, containing 10 examples of each of the 7 classes contained in the original dataset (6 classes of ICDs for other urinary tract disorders and 1 class including examples of different diagnoses). It should be noted that, according to Table 2, there are no examples for the N39.2 code. This measure will also enable us to analyze how well a classifier trained with a highly imbalanced dataset performs on a balanced test set.

These artificial examples were also created using the GloVe algorithm. This algorithm was used to create word embeddings for the input layer. Once the algorithm is trained with the text in the discharge summaries, it is possible to verify words similar to the target word by calculating the cosine similarity between word-vectors. Thus, for each real word contained in a randomly chosen discharge summary, we made the search for the most similar word using GloVe. Once that word is found, we then replace the real word in the augmented test set by the one with most similarity. With this strategy, we intend so simulate different ways of writing the same thing, like different people writing the same concepts.

We evaluate the results according to the tables on section VIII. We classify as our "best model", the one with highest Average F-score (Avg. F-score), lowest Standard deviation (Std. deviation) and higher accuracy on the augmented test set (% Acc. Aug). For the best overall model we also present the F-score for each of the classes, as well as a confusion matrix for the augmented test set.

8. RESULTS

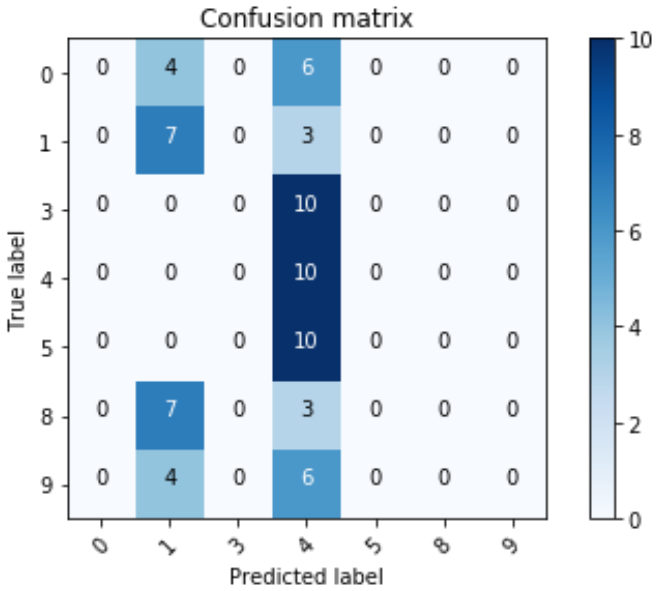
After preprocessing, our database contained 327.529 tokens, with 20.333 distinct words. By implementing the 1-CNN baseline model (Kim, 2014) suggested by Zhang and Wallace (2016), we obtained bad results, as demonstrated on Table 3.

Table 3: Results for the baseline method

Model	Convolution Layers	Window Size	Avg. F-score	Std. deviation	% Acc. Aug.
Baseline	1	3,4,5	0.08	0.05	11%

A further look into the classification accuracy for the baseline model shows us that it was not able to generalize well enough, since mostly all predicted examples were set in Class 4 (Figure 4).

Figure 4: The baseline model implemented on our dataset



We then expanded the baseline model to check which configuration would best fit our problem scenario. Results of the trained CNNs are divided by embedding size and the best models for each embedding size are shown in plots, while the best model overall is further explored by analyzing its confusion matrix as well.

Table 4: Results for the models trained with embedding of 50 dimensions

Model #	Convolution Layers	Window Size	Avg. F-score	Std. deviation	% Acc. Aug.
1	1	5	0.86	0.15	70%
2	2	5,8	0.91	0.13	65%
3	3	5,8,10	0.95	0.05	71%
4	4	5,8,10,12	0.93	0.07	75%
5	5	5,8,10,12,15	0.95	0.06	81%

Table 5: Results for the models trained with embedding of 100 dimensions

Model #	Convolution Layers	Window Size	Avg. F-score	Std. deviation	% Acc. Aug.
6	1	5	0.88	0.15	66%
7	2	5,8	0.92	0.12	77%
8	3	5,8,10	0.95	0.07	82%
9	4	5,8,10,12	0.88	0.03	77%

10	5	5,8,10,12,15	0.81	0.07	63%

Table 6: Results for the models trained with embedding of 300 dimensions

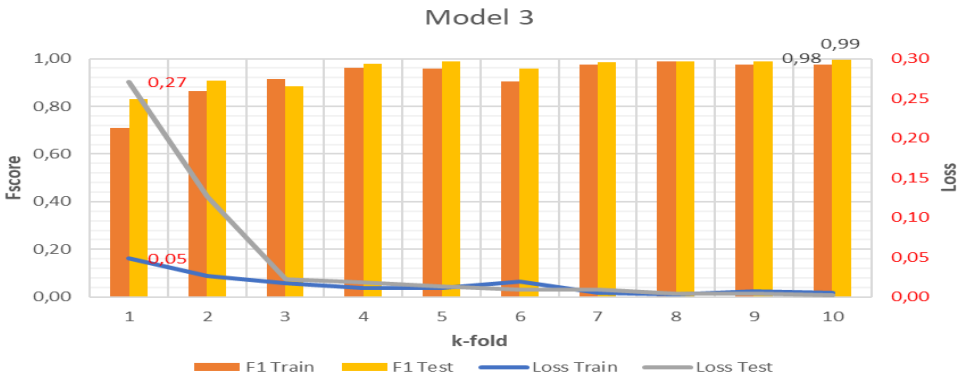
Model #	Convolution Layers	Window Size	Avg. F-score	Std. deviation	% Acc. Aug.
11	1	5	0.92	0.08	64%
12	2	5,8	0.95	0.05	77%
13	3	5,8,10	0.94	0.04	78%
14	4	5,8,10,12	0.94	0.08	63%
15	5	5,8,10,12,15	0.94	0.02	80%

Table 7: Results for the models trained with embedding of 500 dimensions

Model #	Convolution Layers	Window Size	Avg. F-score	Std. deviation	% Acc. Aug.
16	1	5	0.91	0.13	76%
17	2	5,8	0.90	0.10	66%
18	3	5,8,10	0.95	0.09	76%
19	4	5,8,10,12	0.97	0.04	83%
20	5	5,8,10,12,15	0.96	0.05	82%

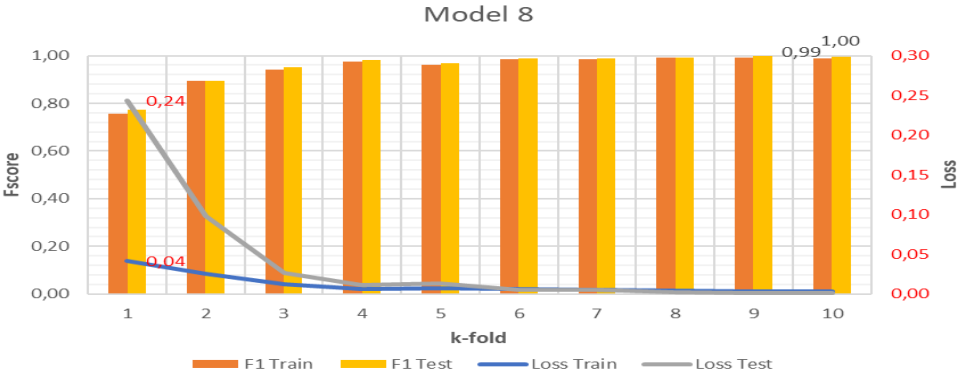
For the 50-dimensional embeddings, it was verified that the model with 3 layers of convolution (Figure 5) have higher average F-score during training and test, also with the lowest values of cost function, demonstrating the lack of overfitting. However, this model wasn't the best on classifying the augmented instances.

Figure 5: Model#3 starts training with significant differences in train and test in terms of F-score and Loss, however, this difference starts to lower down as the training continues until it stays almost even at the end



For the results with embeddings constituted of 100 dimensions, the model with three convolutional layers (Figure 6) presented good performance, and has no apparent overfitting pattern.

Figure 6: Model#8 is, on our understanding, the best model for 100-dimensional embeddings because it has high average F-score, relatively low standard deviation and high accuracy on augmented test set



For the models made from 300-dimensional embeddings, the model with five convolutional layers (Figure 7) presented the best results, with no evidence of overfitting.

Figure 7: Model#15 is the most complex in terms of number of convolutional layers for its embedding size

Finally, we verified that the model with 4 convolutional layers (Figure 8) performed better on training, test and test augmented, being our best model overall. For the sake of brevity, only for this model we present the plot with the results for each class on the artificial test set.

Figure 8: Model#19 is, based on our criteria, the best model on our experimentation

Table 7: Class Statistics for the best overall model (Model #19)

Class	Precision	Recall	F-score
0	0.59	1.00	0.74
1	1.00	0.80	0.88
3	1.00	0.90	0.95

4	0.90	0.90	0.90
5	0.71	1.00	0.83
8	1.00	0.90	0.95
9	1.00	0.30	0.46

Figure 9: Confusion Matrix for Model#19 demonstrated how well it performed on the balanced test set

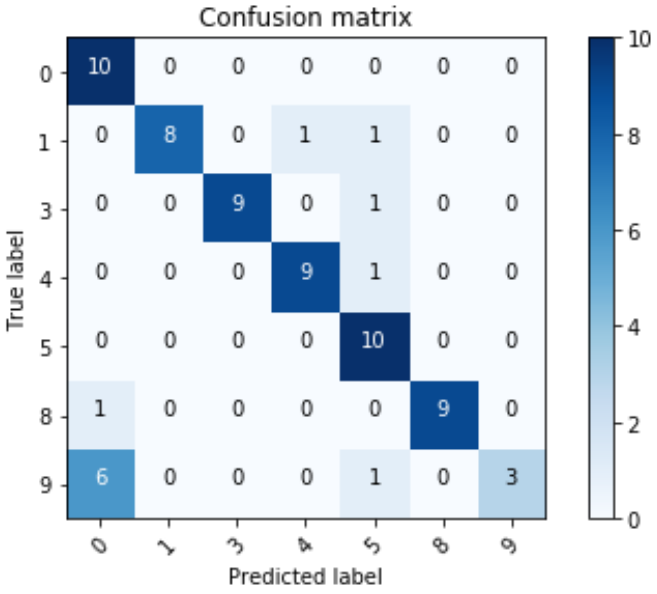
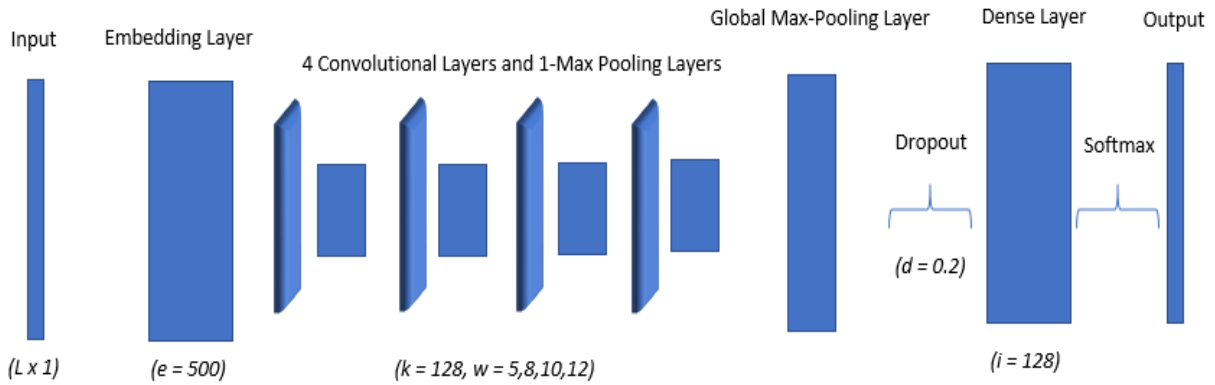


Figure 10: Represents our best model as well as the adaptations we done to make it the best fit for clinical coding



9. CONCLUSION

The results allow to verify that our model outperforms the F-score obtained in Ayyar and Walk (2016) and in Rios and Kavuluru (2015) for the task of clinical text classification. Our models generalize well, with no evidences of overfitting. Additionally, models with a high average F-score and with a low standard deviation tend to have better generalization capacity, evaluated from the F-score in the test set in each fold and the number of instances correctly classified in the augmented set dataset created for balanced testing. In this way, the best results were identified in model 19, suggesting that larger embeddings with more convolutional layers capture better features for classification in relatively small dataset. We also noticed that the self-taught word embeddings represent a viable method for generating context specific inputs for a Deep Learning classifier. It was possible to verify excellent results in all sizes of embeddings, with the most varied model configurations. However, it is possible to verify that models with less convolutional layers tend to have smaller F-score and greater errors in the classification of the artificial instances.

Important to notice that the number of layers seem to influence the model's generalization capacity, as well as the execution time. Models with more layers seem to have better average F-score if compared to the ones with low number of convolutional layers. From this approach, it seems fair that intermediate models (with 3 convolutional layers, for example) generalize well enough, but not being the best.

It is also important to notice that the dropout, along with early-stopping quickly decrease the effects of the overfitting verified in the first folds in training, so that the loss in training and in the test, are equivalent to the end of the training of almost all the models. Another interesting finding of this research is that even the best model classify with low assertiveness class 9, which represents the ICD code N39.9 (Unspecified disorders of the urinary tract), misclassifying these examples by classifying it as N39.0 (Urinary tract infection of unspecified location). We estimate that this result is due to the fact that this class represents a very general class, which context is linked to urinary tract infection of unspecified location, so it is difficult for the classifier to decide what code assign for this type of diagnosis. It is also possible to verify that, although there is a severe imbalance of the dataset, the classifier performed well on the balanced artificial dataset used for further evaluation of the model. This indicates that cost-sensitive learning was effective to treat imbalance.

In general, it is verified that the model proposed in this research points to the possibility of effective automation of the coding of clinical coding using DL with self-taught word-embeddings as input of the model.

10. FUTURE WORK

The positive results demonstrated that it may be interesting to expand this research to other clinical texts or ICD-codes, aiming to automate the clinical coding process for all codes available. Additionally, since our model aims to classify only one group of ICD-10 code (N39), it might be necessary to expand to other examples from other sources to ensure the approach proposed in this research generalizes well for examples not seem in this database.

REFERENCES

[1] PREDA, A. L.; CHIRIAC, N. D.; MUȘAT, S. N. Aspects of clinical coding. **Management in health**, v. 16, n. 3, p. 12-21, 2012.

[2] AALSETH, P. **Medical Coding: What It Is and How it Works**. Albuquerque: Jones & Bartlett Learning, 2006.

[3] WORLD HEALTH ORGANIZATION (WHO). ICD-11 Update. **Health Data Standards and Informatics**, p. 11-14, 2017.

[4] LAURENTI, R. Análise da informação em saúde: 1893-1993, cem anos da Classificação Internacional de Doenças. **Revista de Saúde Pública**, v. 25, p. 407-417, 1991.

[5] STANFILL, M. H. et al. A systematic literature review of automated clinical coding and classification systems. **Journal of the American Medical Informatics Association**, v. 17, n. 6, p. 646-651, 2010.

[6] ZHANG, X.; LECUN, Y. Text understanding from scratch. Computer Science Department, Courant Institute of Mathematical Science, p. 1-10, 2016.

[7] PACHECO, E. J.; NOHAMA, P.; SCHULZ, S. Codificação de narrativas clínicas para uma ontologia de domínio. **Revista Brasileira de Pesquisa em Saúde/Brazilian Journal of Health Research**, v. 15, n. 2, p. 94-103, 2013.

[8] HUANG, F. et al. Learning representations for weakly supervised natural language processing tasks. **Computational Linguistics**, v. 40, n. 1, p. 85-120, 2014.

[9] COLLOBERT, R.; WESTON, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: **Proceedings of the 25th international conference on Machine learning**, 2008, p. 160-167.

[10] SANTOS, A. B. V.; CARVALHO, D. R. Deep Learning for HealthCare Management and Diagnosis. **Iberoamerican Journal of Applied Computing**, v. 5, n. 2,p. 15-25, 2016.

[11] PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. 2014. p. 1532-1543.

[12] LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, v. 521, n. 7553, p. 436, 2015.

[13] SANTOS, C.; GATTI, M. Deep convolutional neural networks for sentiment analysis of short texts. In: **Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers**. 2014. p. 69-78.

[14] HERMANN, K. M. et al. Teaching machines to read and comprehend. In: **Advances in Neural Information Processing Systems**. 2015. p. 1693-1701.

[15] LOPES, F. Actividade dos Codificadores Deve Ser Reconhecida pela Ordem. **Revista Norte Médico**, pp. 10-12, 2009.

[16] CHU, S. Information Retrieval and health/clinical management. **Yearbook of Medical Informatics**, v. 11, n. 01, p. 271-275, 2002.

[17] HARRIS, Z. S. Distributional structure. **Word**, v. 10, n. 2-3, p. 146-162, 1954.

[18] RUBENSTEIN, H.; GOODENOUGH, J. B. Contextual correlates of synonymy. **Communications of the ACM**, v. 8, n. 10, p. 627-633, 1965.

[19] HEUER, H. Text comparison using word vector representations and dimensionality reduction. In: **Proceedings of the 8th European Conference on Python in Science**. 2016. p. 13-16.

[20] MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: **Advances in neural information processing systems**. 2013. p. 3111-3119.

[21] PEREIRA, F. et al. A comparative evaluation of off-the-shelf distributed semantic representations for modelling behavioural data. **Cognitive neuropsychology**, v. 33, n. 3-4, p. 175-190, 2016.

[22] RODRÍGUEZ, I. Text similarity by using GloVe word vector representations. p. 10-17, 2016.

[23] DHINGRA, B. et al. A comparative study of word embeddings for reading comprehension. School of Computer Science, Carnegie Mellon University, n. 3, 2017.

[24] BERARDI, G.; ESULI, A.; MARCHEGGIANI, D. Word Embeddings Go to Italy: A Comparison of Models and Training Datasets. In: **CEUR Workshop Proceedings**. 2015. p. 1-8.

[25] KANG, Hong Jin et al. A Comparison of Word Embeddings for English and Cross-Lingual Chinese Word Sense Disambiguation. Singapure. p. 30-39, 2016.

[26] LV, Xinbo et al. Clinical relation extraction with deep learning. **IJHIT**, v. 9, n. 7, p. 237-248, 2016.

[27] AYYAR, S.; WALK, O. B. D. Tagging Patient Notes with ICD-9 Codes. In: **Proceedings of the 29th Conference on Neural Information Processing Systems**. 2016.p. 1-8.

[28] HUGHES, M. et al. Medical text classification using convolutional neural networks. **Stud Health Technol Inform**, v. 235, p. 246-250, 2017.

[29] RIOS, A.; KAVULURU, R. Convolutional neural networks for biomedical text classification: application in indexing biomedical articles. In: **Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics**. 2015. p. 258-267.

[30] LENC, Ladislav; KRÁL, Pavel. Deep neural networks for Czech multi-label document classification. In: **International Conference on Intelligent Text Processing and Computational Linguistics**. 2017. p. 460-471.

[31] ASPERTI, Andrea; MASTRONARDO, Claudio. The Effectiveness of data augmentation for detection of gastrointestinal diseases from endoscopical images. **Bioimaging**, p. 1-7, 2017.

[32] SUBRAMANIAN, Jyothi; SIMON, Richard. Overfitting in prediction models–is it a problem only in high dimensions?. **Contemporary clinical trials**, v. 36, n. 2, p. 636-641, 2013.

[33] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.. **Deep learning**. Cambridge: MIT press, 2016.

[34] SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, v. 15, n. 1, p. 1929-1958, 2014.

[35] ZHANG, Y.; WALLACE, B. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. **Neural and Evolutionary Computting**, n. 1, , 2016.

[36] HUANG, C. et al. Learning deep representation for imbalanced classification. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. 2016. p. 5375-5384.

[37] KINGMA, D. P.; BA, L. J. Adam: A method for stochastic optimization. In: **Internacional Confeence for Learning Representations**. 2015. p.434-449.

[38] LOUGHREY, J.; CUNNINGHAM, P. **Using early-stopping to avoid overfitting in wrapper-based feature selection employing stochastic search**. Trinity College Dublin, Department of Computer Science, p. 1-6, 2005.

[39] MOU, L. et al. How transferable are neural networks in nlp applications?. In: **Empirical Methods in Natural Language Processing**. 2016. p. 479-489.

[40] SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, v. 45, n. 4, p. 427-437, 2009.

[41] WANG, W. Y.; YANG, D. That's So Annoying!!!: A Lexical and Frame-Semantic Embedding Based Data Augmentation Approach to Automatic Categorization of Annoying Behaviors using# petpeeve Tweets. In: **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**. 2015. p. 2557-2563.

[42] SAKHAVI, S.; GUAN, C.; YAN, S. Learning Temporal Information for Brain-Computer Interface Using Convolutional Neural Networks. **IEEE Transactions on Neural Networks and Learning Systems**, v.29, n. 11, p-5619-5629, 2018.

[43] JIANG, B. et al. Latent Topic Text Representation Learning on Statistical Manifolds. **IEEE Transactions on Neural Networks and Learning Systems**, v.29, n. 11, p. 5643-5654, 2018.

[44] PANG, Y. et al. Convolution in convolution for network in network. **IEEE transactions on neural networks and learning systems**, v. 29, n. 5, p. 1587-1596, 2018.

[45] LI, P.; HUANG, H. Clinical information extraction via convolutional neural network. University of Texas in Arlington, p. 1-5 , 2016.

[46] KIM, Y. Convolutional Neural Network for Sentence Classification: In: **Proceedings of the 2014 Conference on Empiral Methods for Natural Language Processing**. 2014. p. 1746-1751.

[47] GANGANWAR, V. An overview of classification algorithms for imbalanced datasets. **International Journal of Emerging Technology and Advanced Engineering**, v. 2, n. 4, p. 42-47, 2012.