Politecnico di Torino

Master's Thesis

Master of Science in
Communication & Computer Networks Engineering

Dilated Convolution Networks for Classification of ICD-9 based Clinical Summaries

Supervisor:

Prof. Maurizio Morisio

Candidate:

Neel Kanwal

Internship Tutor:

Dott. Giuseppe Rizzo (LINKS Foundation)

March 2020

Keywords

Electronic Health Records (EHR), Clinical Notes, Convolution Neural Networks, Artificial Intelligence, Attention Mechanism, MIMIC Dataset, Global Vectors, Word Embedding, Dilated Convolutions, Machine Learning, Medical Records, Text Processing, International Classification of Diseases (ICD-9), Distributed Models.

Abstract

Deployment of Artificial Intelligence for understanding clinical notes in the healthcare sector is a crucial step to extract meaningful phrases based on diseases. Electronic Health Records (EHR) are stored in the health care system in an unstructured and event associated way. Public clinical records can be used for billing, monitoring and insurance purpose. These clinical notes contain abbreviations, acronyms, and a non-uniform dictionary. Various Machine learning models are used with different approaches to understand these notes, these models are evaluated in various criteria based on datasets. These techniques differed mainly in pre-processing and code assignments as well as architecture for reading long medical documents.

In this thesis work, we propose a layered model of Convolution Neural Networks with pre-trained embeddings. This architecture uses multiple dilation layers with a label-specific dot-based attention mechanism. We have extracted the embeddings from Common Crawl GloVe (Global Vector). The architecture of the model is designed to calculate attention to words and their context. The model is trained on MIMIC-III data set labeled with the ICD-9-CM hierarchy.

This research is helpful in highlighting and perceiving useful information from medical reports to a physician, this step will apparently increase treatment quality and support administrative tasks. We conclude with optimal results compared to state-of-the-art models, proposing certain limitations and possible developments in the near future. It will address a significant role in health security for nations using their public healthcare systems.

Acknowledgements

I would like to thank Dott.Giuseppe Rizzo (LINKS Foundation) for being available all the time to provide valuable feedback. His directions and supervision facilitated me learning beyond the scope. I would like to appreciate Mr. Stefano Malacrino (Institute of Bio-Medical Engineering, University of Oxford) for helping in technical stuff and providing evaluations based on his prior research experience of the similar domain. His previous work was a huge support to carry on my further research.

I am affirming gratefulness to my supervisor, Prof. Maurizio Morisio for helping in availing resources necessary to carry on my research throughout the entire period.

Contents

List of Figures

List of Tables

List of Acronyms

CNN Convolution Neural Network

MIMIC Medical Information Mart for Intensive Care

ICD International Classification of Diseases

EHR Electronic Health Records

HIPPA Health Insurance Portability and Accountability Act

PR-AUC Precision Area Under the Curve

ROC Region of Convergence

NLP Natural Language Processing

BOW Bag Of Words

TPR True Positive Rate

FPR False Positive Rate

SVM Support Vector Machine

CBOW Continuous Bag Of Words

TFIDF Term Frequency Inverse Document Frequency

GloVe Global Vectors

RL Reinforcement Learning

LSTM Long Short-term Memory

RNN Recurrent Neural Networks

GPU Graphic Processing Unit

ANN Artificial Neural Network

HA-GRU Hierarchical Attention Gated Recurrent Unit

CAML Convolution Attention for Multi-Label Classification

BERT Bidirectional Encoder Representation from Transformer

OOV Out of Vocabulary

CHAPTER 1

Introduction

1.1 Overview

Technological advancement can be observed in every field nowadays. Today, the empirical research and experiments are results of common phenomena, some of them are unexplainable and different from previous trends. New technological resolution to these phenomena is more adaptable and cognitive now than they used to be decades ago. High speeding processing, memory performance and electronic bottlenecks have been widening at a much higher speed. This solution produces space for high efficient algorithms and artificial intelligence technologies to grow.

Human communications nowadays are more text-based because of social media. Natural language can be utilized by many algorithms to extract meaningful understanding. The processing of this natural language for human-computer interaction is known as Natural language processing(NLP). It is used to understand commonly used language and feed it to algorithms. The involvement of NLP is generalized into two main streams, Natural Language Understanding (NLU) and Natural Language Generation (NLG).

After the advent of social media networks and verbal enriched platforms, the need for developing text-based algorithms has given broader space. Every document presents a topic as a general context of text. This may help to predict much about the situation. In theory, we can understand and even predict human behavior using that information.

1.2 Health Records

The healthcare sector is one of the most important public care departments in any country. It monitors national health and governs the public health of the coming generations. Electronic Health Records (EHR) contains patient's basic information like date of birth, height, weight, blood-pressure, blood-sugar level and other suggestions to doctors [26]. In some cases when this history becomes chronic, we have a lot of text related to prescriptions, medical procedures, and discharge suggestions. EHR frameworks are supposed to keep track of information precisely in a timestamped manner. It helps to avoid the use of any allergic drug or practice in the future. Substantially saving billions of dollars for any economy worldwide.

With the rising popularity of clinical notes for text-processing, National central bodies started making it safer for research purposes. HIPPA (Health Insurance Portability and Accountability Act) has allowed medical institutes and research organizations to avail publicly de-identified patient records [45]. These notes contain no personal data and assign Subject-ID to every person for their relevant profile. Moreover, these documents are labeled with the International Classification of Diseases (ICD) standard to ensure the proper multi-label classification of several records and identify diseases [28].

1.3 Goal

The purpose of this work is to develop an architecture that can understand clinical notes and their ICD labels. The available notes are very vague and contain a lot of unnecessary details. Data is processed prior to feeding to a Machine Learning model. This research work in also an academic requirement of masters degree program. This practice will facilitate in developing research-based knowledge of real-world problems in a sophisticated manner. In this document, we will try to utilize the existing state-of-the-art frameworks and develop deep five-layer dilation of neural networks with various embeddings and stacking those embeddings in a supplementary fashion to cop with out-of-vocabulary works and to tune the results and achieve more efficient framework.

The main objective is to predict labels based on text. Results will be described in several evaluation criteria. The trained model will provide a huge application of automated disease labeling for physicians based on available notes. It can further post-process the text and highlight the important phrases out of it.

CHAPTER 2

Related Work

Text classification is a problem where we have a fixed set of classes and any given text is assigned to one of these categories. In clinical documents, these classes are labeled, proclaiming a particular disease of a patient. This can be done with numerous machine learning approaches.

2.1 Basic Definitions

Artificial Intelligence is an applicable field of machine learning where trained models are deployed to make a decision and mimic human behavior based on their training. In the real world, we have our experiences and prior knowledge of the particular areas to make a decision. Similarly, computers are trained and tested on data-sets to make proper judgments. These Machine learning algorithms are usually categorized into three, supervised, unsupervised and reinforcement learning.

2.1.1 Supervised Learning

In this method, data has a described relation between ground truth output and input. This means data is labeled: for example if we have several text-documents and we know their genre we can predict the similar document relation for others. This approach is widely used in sentiment analysis, text classification and spam filtering. Some most common algorithms are Support Vector Machine, Naïve-Based and Nearest Neighbour for classification and Linear Regression, Decision Trees, Neural Networks for Regression [47].

2.1.2 Unsupervised Learning

Contrary to supervised learning, data is unlabelled here which defines the main objective of arranging data in structures and identify patterns. Unsupervised Machine Learning has three basic objectives, dimensional reduction, Clustering, and Association. This technique is simply used in Sentence segmentation, Machine Translation, and Dependency Parsing. Some common algorithms are fuzzy logic, Bayesian Clustering, Hidden Markov Model, PCA (Principal Component Analysis and LDA (Linear Discriminant Analysis) [47].

2.1.3 Semi-Supervised Learning

Semi-supervised learning techniques are a blend of the previous two (supervised and unsupervised) exhibited in the above section, this approach address problem where majority samples of the training are unlabeled, even though only limited data points with label are available. Advantage of this is that in several areas a huge amount of unlabeled data points is willingly available. Applications, where semi-supervised learning is used, are nearly the same as supervised learning [47].

This type of learning is most beneficial when the labeled data points we have are not too common or so exclusive to get then using that unlabeled available data points can raise the performance. It has a common application of speech analysis and web-content classification.

2.1.4 Reinforcement Learning

This approach is widely used in robotics, record management, and finance where the prime goal is to develop a policy. For instance: in games where a correct step gives some rewards and wrong movement penalize the score. The Reinforcement Learning helps agents to learn by witnessing the available behaviors and their conduct by using only an evaluative response, called the return. The policy's ultimate goal is to increase its long-term success. Few well-known algorithms are Q-learning, SARSA, Deep-Q Network [48].

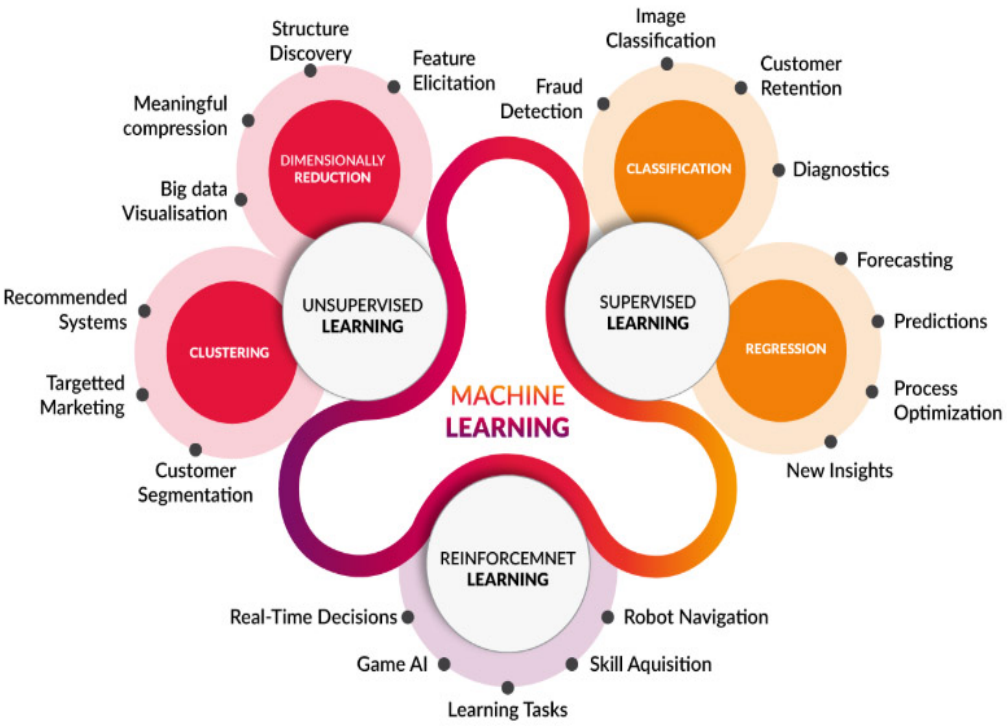

Figure 2.1: Description of Machine Learning types, taken from https://cleanpng.com/Rusbert

2.2 Fundamentals of Text-Processing

Text processing is a staging process, which requires filtering text according to an end-to-end pipeline. This pipeline comprises tokenizing raw text, cleaning punctual and other signs, vectorizing text, applying the desired model for training, validating model and filtering desired results.

2.2.1 Tokenization

Tokenization is a very useful process before applying NLP methods. It is simply breaking a stream of text into words, phrases or meaningful symbols knows as tokens. Many built-in methods are used for splitting sentences of clinical notes based on spaces, punctuations, medical notations and symbols. It is also known as lexical analysis.

2.2.2 Text Cleaning

After the text is split, Some most commonly repeated words which are known as stopwords are removed. These tokens do not contain much contextual meaning and are often repeated for grammatical purposes. Stemming and Lemmatization are known as text normalization techniques. Stemming removes suffixes, prefixes and extra additions to words. These words are normally inflected with addition to the present form of the verb.

Lemmatization is similar to stemming but it brings context to the words. So it links words with similar meaning to one word. Text preprocessing includes both Stemming as well as Lemmatization. Many times people find these two terms confusing. Some treat these two as same. Actually, lemmatization is preferred over Stemming because lemmatization does a morphological analysis of the words.

2.2.3 Vectorizing Text

Machines can only understand and process numbers but texts. So these cleaned tokens need to be converted in vectors of number in a reasonable way. This process of converting text into numbers is known as vectorization or embeddings. There various techniques available for this task.

One-hot Encoding

Integer encoding or frequency-based encoding suffers from many problems such as long-tail or distributional bias. Some token may appear very regularly whereas some may be less common. This creates a need to present words in a one-dimensional vector. This form of vectorization takes categorical data and return numerical binary data for it. Here (1) shows the presence and (0) poses the absence.

For Example:

“Black” ===> [1,0,0,0]

“White” ===> [0,1,0,0]

“RED” ===> [0,0,1,0]

“BLUE” ===> [0,0,0,1]

0ne-hot encoding eliminated the tokens distribution disparity and is most appropriate for shorter documents with fewer repetitive words.

Bag-of Words Model

Bag-of-Words is the crudest model available in Natural Language Processing. It predicts the current word based on context. This representation is formed based on the occurrence of words in a document. It involves a vocabulary of knows words and the total count of the presence of known words. Order of words or information is lost that's why it is called a "bag".

For Example: "It is the best time. It is the time when we have the technology."

Here we have 9 unique words. We will make a frequency vector with one-position for the corresponding word.

"It" ===>[1 0 0 0 0 0 0 0 0]
"is" ===>[0 1 0 0 0 0 0 0 0]
"the" ===>[0 0 1 0 0 0 0 0 0]
"best" ===>[0 0 0 1 0 0 0 0 0]
"time" ===>[0 0 0 0 1 0 0 0 0]
"when" ===>[0 0 0 0 0 1 0 0 0]
"we" ===>[0 0 0 0 0 0 1 0 0]
"have" ===>[0 0 0 0 0 0 0 1 0]
"technology" ===>[0 0 0 0 0 0 0 0 1]

Now to vectorize the sentence , we present the count on corresponding position in frequency vector.
"It is time when we have technology" ==> [1 1 0 0 1 1 1 1 1]

There comes a problem when corpus increases and we have a large vocabulary this results in a vector with a higher dimension and a lot of zeros for less frequent words. It requires memory resources and makes computation inefficient. A different approach of k-gram can be used in this case to combine multiple common recurring words and use on position for them [15].

Term-Frequency (TF-IDF)

A concern with scoring word frequency is that in the text, highly common words tend to dominate (e.g., greater score), but may not contain as much "information content" as rare but perhaps domain-specific words to the model. One solution is to re-scale the frequency of terms by how often they occur in a document, in order to penalize the scores for commonly used words such as "the," which are also common across all documents [46].

TF-IDF takes into account the relative frequency of tokens in the database in other corpus documents against their size. Term Frequency tf(t,d) is the number of times that term t occurs in document d but there are many variants with different weights. Generally, term frequency is scaled logarithmically to prevent bias caused by longer documents or terms that appear much more frequently with respect to other terms: tf(t;d) = 1 + log(ft;d). Inverse document frequency idf(t,D) is the logarithmically scaled inverse fraction of the documents that contain the word obtained by dividing the total number of documents (N) by the number of documents containing the term (n_t), and then taking the logarithm of that quotient $\log(N/n_t)$.

The TF-IDF is the product of two statistics, term frequency, and inverse document frequency. There are various ways of determining the exact values of both statistics.

$$TF-IDF(t,d,D) = tf(t,d).idf(t,D) = \log(N/n_t).\ (1\ +\ log(ft,d))$$

The result of each term lies between [0,1] where term closer to 1 is more meaningful and vice versa.

For Example:
The cat and the dog play ===> [0.42 0.00 0.30 0.42 0.00 0.42 0.60]

Cosine Similarity

Counting the common words or Euclidean distance is the general approach used to compare related documents based on the number of mutual words between the documents. Even if the number of common terms increases, this strategy will not work, but the document talks about different topics. The "Cosine Similarity" technique is used to evaluate the resemblance between the documents to solve this error. Mathematically θ, smaller the angle, higher is the similarity between the words.

$$cos\theta = \frac{\overrightarrow{a.b}}{||a||.||b||}$$

Cosine similarity not only demonstrates vector similarity, but it also avoids word count frequency.

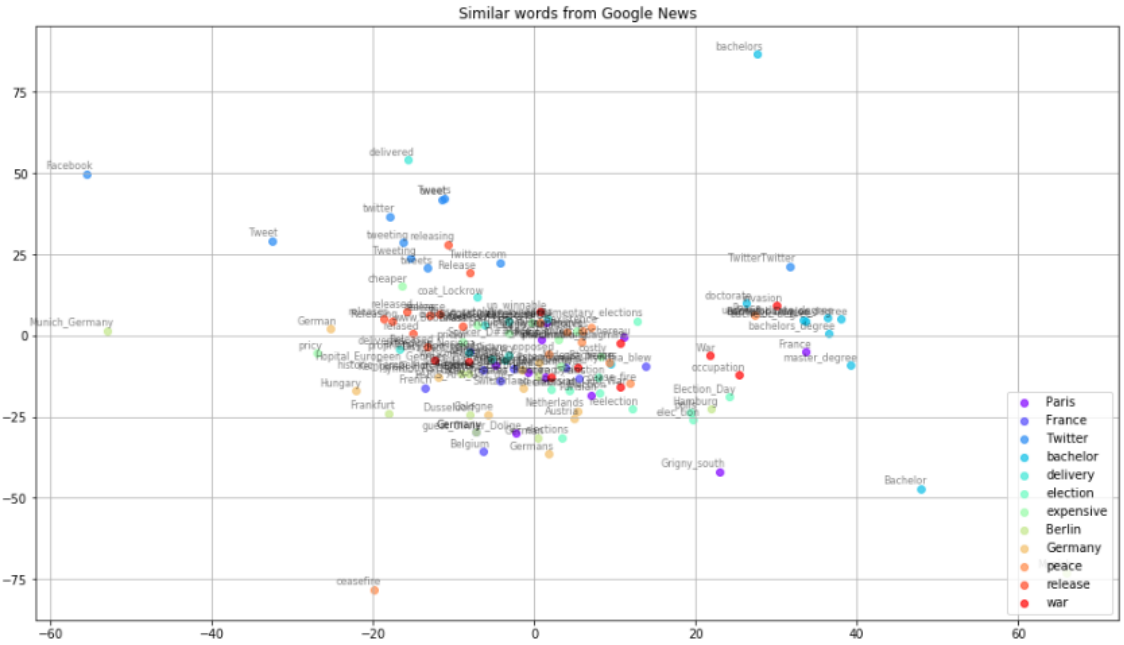

Figure 2.2: Word Similarity From Word2Vec Model

Distributed Models

Words embeddings are efficiently generally formed the pre-trained models for large corpus. It is a faster approach that follows the property of cosine similarity. The most frequent problem in these models is out-of-vocabulary words and the use of language in which similar distribution in language has a similar meaning. These similarities are either paradigmatic like things which co-occur, bee and amp; honey, light, and bulb. And syntagmatic similarities for things that are similar and used to extract some kind of word from a vector. We will describe some of those models here such as Word2vec, GloVe and FastText. Figure 2.2 shows how these models form the space for similar words.

Word2Vec Model

Natural language processing systems treat words as a tiny unit. NLP systems have constraints to calculate this similarity especially when there are billions of tokens. Word2Vec was introduced by Mikolov et al (2013) [15]. It is a deep learning technique where a two-layer neural network takes input and produces a corresponding vector space. Essentially, Word2Vec positions the word in the space of the pre-trained model so that its location is determined by its meaning, i.e. words with similar meanings are grouped together and the distance between two words also has the same meaning. For Example, Words share the same analogy as "Paris is to France as Berlin is to Germany".The vector V_{Paris}, V_{France}, V_{Berlin} and $V_{Germany}$ are encoded in a fashion that it reserved their semantic meaning.

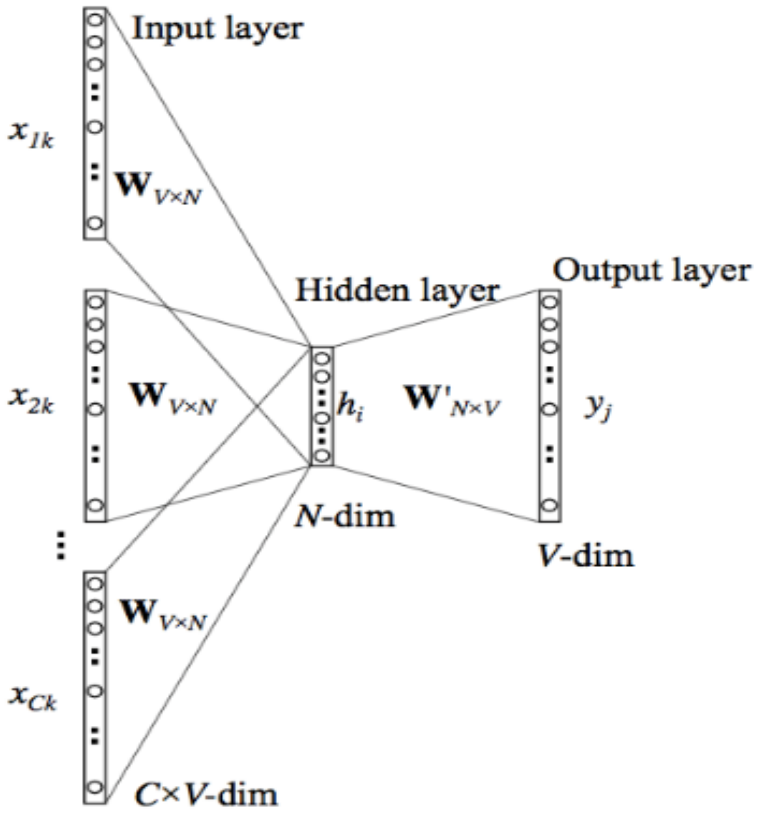

Figure 2.3: CBOW Model from Original Paper[15]

Word2Vec has two main architectures:

Continuous Bag-of-Words(CBOW):

In this model, we find the probability of focused words given the corpus. CBOW is primarily used in smaller datasets since it treats the context of the sentence as a single observation towards predicting the target word. In practice, this becomes very inefficient when working with a large set of words.

In figure 2.3 , Input Layer x is an encoded vector of x = {$x_{1k}, x_{2k}, \ldots, x_{Ck}$} where C is the context window and V represents vocabulary size. All input layer is forwarded to a hidden layer with a particular weight matrix $W \in \mathbb{R}^{V \times N}$ Where N is a number of the hidden layers. Afterward, Hidden layers are few to output layer with weight matrix $W' \in \mathbb{R}^{V \times N}$. This procedure takes in the background window the W rows corresponding to the words vocabulary indexes (this is a function of x's one-hot encoding) and averages them. Later used for computation at the output layer.

$$u_j = hv'_w j$$

$$h = \frac{1}{C}(\sum_{i=1}^{C} X_i)W$$

$$y_j = p(w_y j | w_1, ..., w_C) = \frac{exp(u_j)}{\sum_{j'=1}^{V} exp(u_j')}$$

The final output y_j is computed by passing u_j through the softmax function. Loss Function here is to minimize the conditional probability.

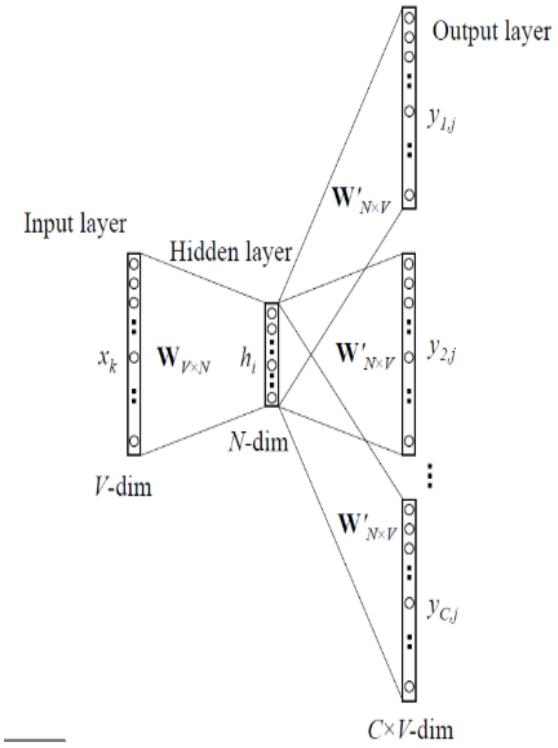

Figure 2.4: Skip-gram Model from Paper [15]

$$\zeta = -\log(p(w_0|w_1))$$

$$\zeta = -u_j * -\log(\sum_{j`=1}^{V} exp(u_j`))$$

Continuous Skip-gram Model: Here we find the probability of the corpus-based on focus words. We use negative sampling here to avoid softmax computation. Spearman correlation is used to compare the ranked list with cosine similarities. There is an extension for Word2Vecfor Bioinformatics. (ProtVec and BioVec).

The Figure 2.4 shows that this is opposite to CBOW as defined before. Here we predict the output from hidden layers. Hidden Layers compute h = x.W and output hidden layer is forwarded as

$$u_{\mathrm{c,j}} = u_{\mathrm{c}} = hv`_{\mathrm{wj}} \forall j \in \{1,2,...,C\}$$

Loss function changes here because of C nominal distribution.

$$\zeta = -\log(p(w_{0,1}, w_{0,2}, ..., w_{0,C}|w_I))$$

$$\zeta = -\sum_{c=1}^{C} u_{c,j} * +C\log(\sum_{j`=1}^{V}) exp(u_j`)$$

Output Layer results:

$$y_{c,j} = p(w_c j = w_{0,\mathrm{c}}|w_I) = \frac{exp(u_{\mathrm{c,j}})}{\sum_{j`=1}^{V} exp(u_{\mathrm{j}`})}$$

FastText

FastText is Facebook's planned expansion of Word2Vec, released in 2016 [40]. FastText splits words into several n-grams (sub-words) instead of feeding individual words into the Neural Network. For example, the tri-grams for the word apple are "app", "ppl", and "pl" (ignoring the beginning and end of the word boundary). The term enclosing vector for apple will be the sum of all these n-grams. Figure 2.5 shows the structural form of FastText.

X_{i} is the n-gram feature that will be converted to (one-hot encoded representation) embedded and averaged from the hidden variable. This design is similar to the paper [15]. CBOW model (2013) with a label that is replacing the middle word. A softmax function is used to calculate probability distribution over a number of classes but when the number of classes is large, computing the linear classifier is computationally expensive [40].

$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(f(BAx_n))$$

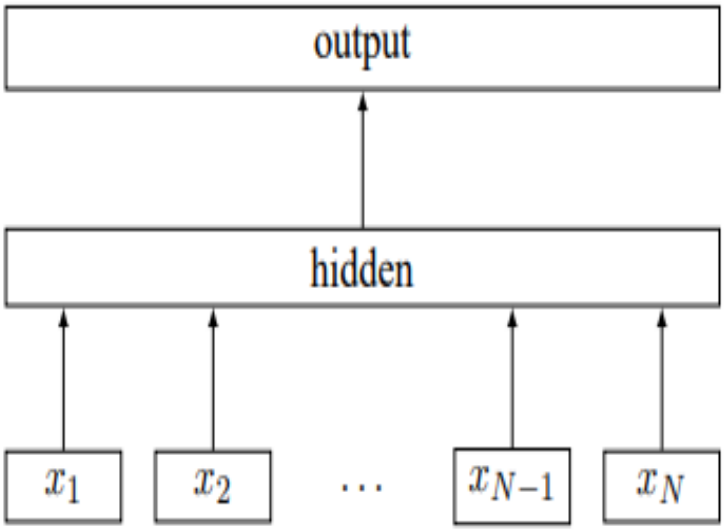

Figure 2.5: FastText Architecture from paper [40]

In the equation above, A is lookup Matrix, B is Output transformation and hierarchical softmax is applied to their product. Often, when looking for the most likely class, the hierarchical softmax is beneficial at test time. The likelihood that the path from the root to this node is associated with each node. So a node is always less probable than the parent's probability.

GloVe (Global Vectors)

Word2vec was mainly based on preserving semantic analogies on basic algorithms using Neural Networks. It was a window-based approach and has the demerit of ignoring global statistics. The GloVe makes a global co-occurrence matrix by approximating the probability a given word will co-occur with other words. This presence of globals GloVe ideally work better. GloVe, 2014 introduced by Pennington et al [5] at Standford was a log-bilinear model combing local context- matrix factorization.

The count matrix in the case of GloVe is preprocessed by standardizing numbers and smoothing them. The GloVe makes the parallel implementation, in comparison to word2vec, making it much easier to train more data. This produces the word vectors, which operate well both on word comparison and on similarity tasks and on the identification of entities. Matrix factorization methods for producing low-dimensional representations of words have origins that stretch back to Latent Semantic Analysis.

The first step is to create a matrix for co-occurrence. Calculation of the matrix with a fixed window dimension (words are jointly valid when they appear together in the same window) takes into account the local context. The GloVe is a count-based model whereas count-based models learn vectors by doing dimensionality reduction on a co-occurrence counts matrix. On the other hand, Word2vec is a predictive model. The principle of GloVe is that the relationships of co-occurrence between two words in a sense are strongly related to their meaning. Instead of the probabilities of themselves, the correct starting point should be the term vector learning with the coexistence likelihood ratios. The most popular model takes the form of the P_{ik} / P_{jk} ratio, which relies on three terms i,j,& k.

Here, w $\in \mathbb{R}$ and the information present the ratio P_{ik}/P_{jk} in the word vector space. P_{ij} indicates the probability of the term j to appear in context with i and can be calculated as

$$F(w_i, w_j, \hat{w_k}) = \frac{P_{ij}}{P_{jk}}$$

$$P_{ij} = \frac{X_{ij}}{X_i}$$

In the equation above X represents co-occurrence matrix and X_i which is the total number of words that appeared in context. X_{ij} denotes i,j th term in the matrix.

F is a function that takes the embedding of the words i,k,j as input. The number of possibilities for F is vast, but by enforcing a few desiderata we can select a unique choice. Since one of GloVe's goals is to create word vectors that express meaning using simple arithmetic (vector addition and subtraction), a function F must be chosen to match this property with the resulting vectors. Considering that vector spaces are essentially linear structures, with vector differences the most natural way of doing so is to restrict our consideration to those F functions which only depend on the difference of the two target terms. The simplest way to do this is to calculate the input to F as the disparity between the compared vectors.

$$F(w_i - w_j, \hat{w_k}) = \frac{P_{ij}}{P_{jk}}$$

In the case of word-word co-occurrence matrices, the distinction between the word and the meaning word is arbitrary and we are free to swap the two functions. In order to do so reliably, we will swap $X \implies X_T \& w \implies \hat{w}$.

$$F((w_i - w_j)^T, \hat{w_k}) = \frac{F(W_i^T \hat{w_k})}{F(W_j^T \hat{w_k}}$$

The ratio can be transformed into a subtraction of probabilities by taking the logarithm of likelihood ratios, and a bias term is for each word to take into consideration the fact that some terms occur more often than others. The product of these operations can later be converted into an equation over a single entry in the co-occurrence matrix.

$$w_i * \hat{w_k} + b_i = \log(P_i k) = \log(X_i k) - \log(X_i)$$

$$w_i^T \hat{w_k} = \log(P_i k) = \log(X_i k) - \log(X_i)$$

this term is independent of k so it can be absorbed into a bias b_i for w_i. Finally, adding an additional bias $\tilde{b_k}$ for $\widetilde{w_k}$ restores the symmetry.

$$w_i^T \hat{w_k} + b_i + \hat{b_k} = log(X_i k)$$

A major drawback to this model is that all co-occurrences are measured equally, even those that rarely or never occur. To avoid this a weighted least squares regression model by introducing a weight function into Objective Function J. Training is aimed at minimizing J.

$$J = \sum_{i,j=1}^{V} f(X_i j)(w_i^T \hat{w}_j + b_i + \hat{b}_j - \log X_i j)^2$$

Here V is the Vocabulary and Weighted function must obey three rules:

- If function f is viewed continuous then it should vanish faster as square-log limit tends to infinity f(0)=0.
- F (x) should be non-decreasing in order to prevent over weighting with unusual co-occurrences.
- For large values of x, f(x) should be relatively small, so that repeated co-occurrences are not over-weight.

$$f(X_i k) = min(1, (\frac{X_{ik}}{x_{max}})^{\alpha})$$

This function slashes the output of exceedingly common word pairs (where $X_{ij} > x_{max}$) and simply returns one. Since this number is always smaller than the total number of matrix entries, the model scales no worse than O($||V||^2$)[5].

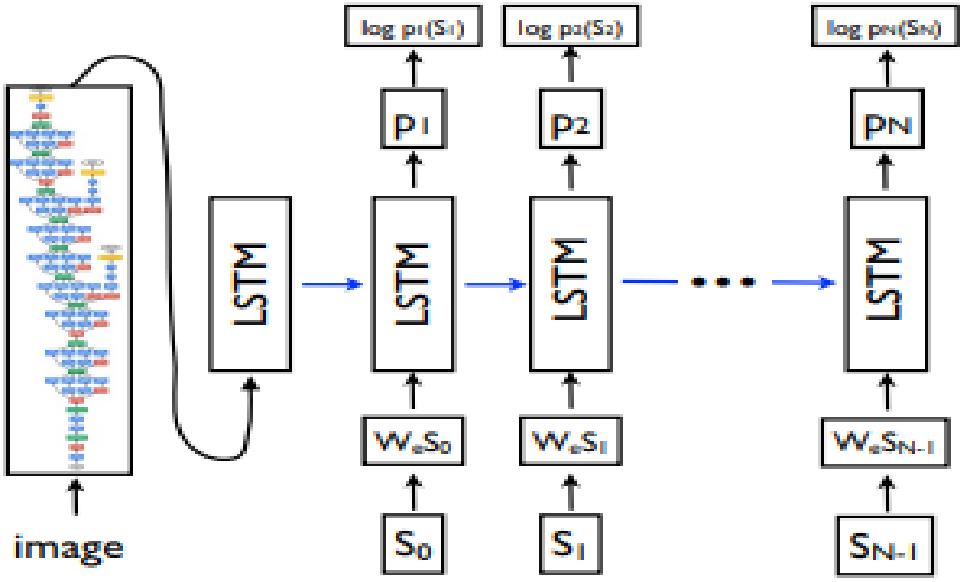

Figure 2.6: Attention Mechanism by Vinalys et al (2015) [44]

2.3 Attention mechanisms

Itti et al (1998) [32] described attention for the first time in his short paper. Attention is similar to visual focus for humans at first sight. This practice was dominantly introduced for computer vision to take

the most important objects from any picture into consideration. The idea of attention-mechanism became popular mainly for its application to image captioning and neuroscience computation in neural network architectures. Cho et al (2014) [31] used it in recurrent neural networks to compute weights of image segments and Vinalys et al (2015) [44] combined LSTM with DeepCNN for image captioning. Figure 2.6 is the model flow for visual attention.

Soft attention is a completely differentiable deterministic mechanism that can be inserted into an existing system, and the gradients are propagated through the process of attention at the same time that they are propagated through the rest of the network. On the other hand, hard attention is a stochastic process in which instead of using all hidden layer weights only a randomly chosen sample is used. Both systems have their advantages and disadvantages, but the trend is to focus on mechanisms of soft attention as the gradient can be calculated directly rather than estimated by a stochastic process. Another similar difference in global attention and local attention described by research work Kelvin Xu et al (2016) [27]. Figure 2.7 describes the generic difference between the two.

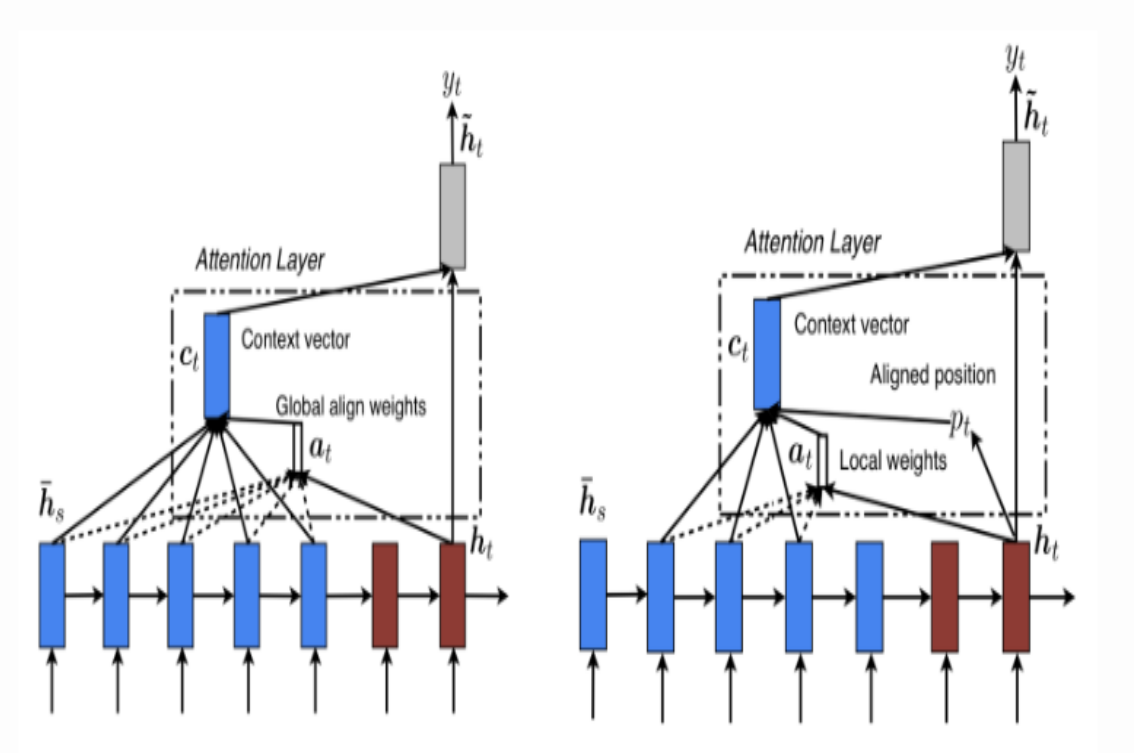

Figure 2.7: Global (L) & Local (R) Attention from paper [25]

In soft attention, all image segments have weights at any given time whereas in hard attention the only patch of the image is considered. But local Attention is not the same as the hard Attention used in the image captioning task. In soft attention, the model is smooth and differentiable but expensive when the source input is large. Whereas in hard attention, less calculation at the inference time and the model is non-differentiable requiring more complicated techniques such as variance reduction or reinforcement learning to train.

Recently, in machine translation attention is gaining popularity because of its encoder-decoder architecture. Transduction model by Bahdanau et al (2014) was first in natural language processing to apply attention in NMT using RNN and CNN [33]. In order to predict or infer a single element such as a pixel or a word in an image or in a paragraph, the attention vector uses the value to determine how strongly it is associated (or "attends to" as you may have read in many documents) with other elements and how much their values are measured by the attention vector. This mechanism is analogous to the human sight where specific spots or words are read at contact. It is commonly said that attention saves time for humans but loses for machines [21].

Self-attention, also called intra-attention, is a process of attention that relates different positions of the individual sequence to determine the same sequence of the representation. The machine interpretation, abstract summarizing or image description creation have proven to be very realistic. Mathematically, three separate vectors, namely key, query, and value, should be used per embedding of the word Such vectors can easily be generated by multiplying matrices (K, V, Q). First, a function f is used to compute the similarity between each key K_i and query Q to obtain weight W. Many different kinds of functions are listed in table 2.1.

Name	**Function**
Content-based Attention	$f(Q_T, K_i) = cosine[Q_T, K_i]$
Additive	$f(Q_T, K_i) = V_a^T * \tanh(W[Q_T, K_i])$
Location-Based	$\alpha_{l,i} = Softmax(WQ^T)$
General	$f(Q_T, K_i) = Q^T W K_i$
Dot-Product	$f(Q_T, K_i) = Q^T K_i$
Scaled Dot-Product	$f(Q_T, K_i) = \frac{Q^T K_i}{\sqrt{n}}$

Table 2.1: Functions for Various Attentions

A softmax is used to normalize the weights and then finally attention weights are used to compute a final context representation

$$a_i = Softmax(f(Q, K_i)) = \frac{exp(f(Q, K_i))}{\sum_i exp(f(Q, K_i))}$$

$$Attention(Q, K, V) = \sum_i a_i * V_i$$

The transformer model is one of the efficient encoder-decoder architectures, presented a lot of improvements to the soft attention and make it possible to do transduction modelling without recurrent network units [2]. The Multihead Mechanism goes through the scaled dot Product attention several times in parallel, rather than only once measuring the attention. Simply connected and linearly translate independent attention outputs into the necessary dimensions. The use of this kind of mechanism is suggested in chapter 8 for future work.

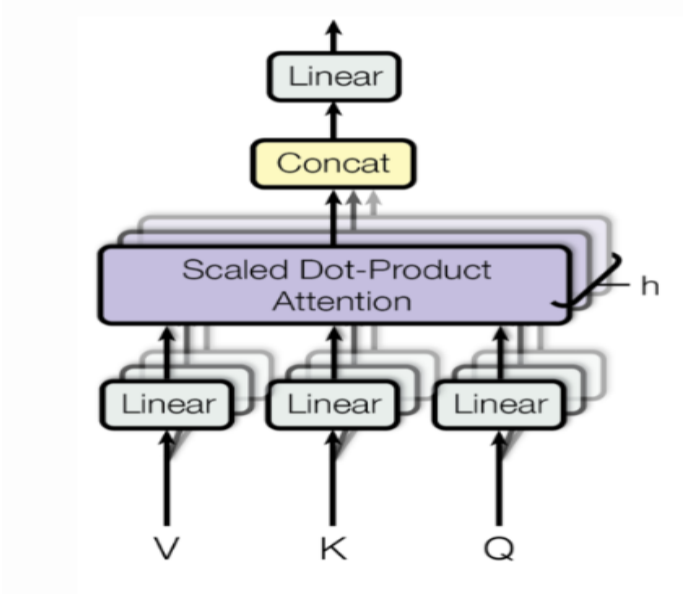

Figure 2.8: Multihead Attention

The documents are created in a hierarchal way where every document contains sentence and sentences are further formed by words and characters. The model proposed by Yang et al 2016, discussed in the literature review where the flow of information is not uniform across the whole document. The attention used in this work is hierarchal in a way that during the layer of dilation we are using different filters to form an n-gram while higher to lower representation [18].

We normalize the word weight by the weight of the sentence because of its hierarchical structure to ensure that only important words are emphasized in important sentences. Here every line is a sentence which sometimes surpasses many lines. Figure 2.9 shows sentence-level attention for semantic analysis a blue depicts word-level attention for meaningfulness. It can be seen from the yahoo answer which words (web, browser, etc) typically contribute more to the question paradigm.

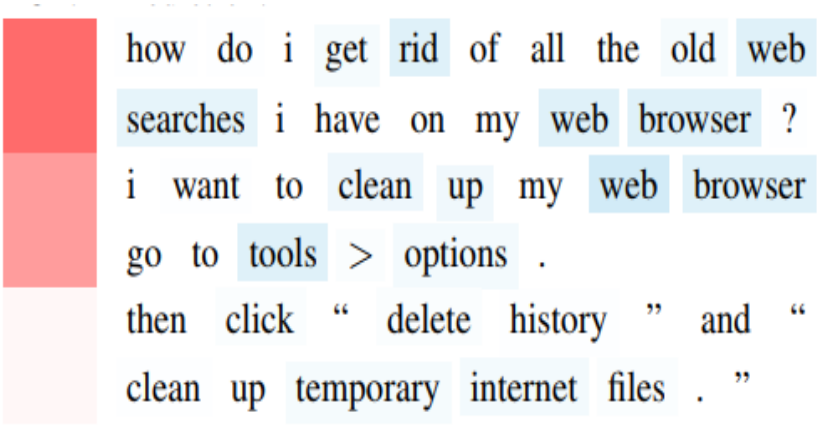

Figure 2.9: Document form Yahoo Answers

2.4 Convolution Neural Networks

Neural networks are composed of human brain-inspired algorithms. Generally, when you open your eyes, what you see is called data and is processed in your brain by the Neurons (data processing cells) and knows what's around you. Alex Waibel et al (1987) [42] introduced convolution networks inspired by the visual cortex of the human brain . Applicative CNN by Yan LeCun et al (1998) were class of deep, feed-forward artificial neural networks used for banking systems to recognize numbers on cheques [41].

Neural Networks are commonly known as Artificial Neural networks (ANN), because of their artificial capacity to mimic brain were primarily used in computer vision tasks. Artificial Neural Networks are universal approximators which means they can form any function. Nevertheless, most recently, Convolutional Neural Networks have found prominence in addressing NLP-related issues such as Sentence Classification, Text Identification, Sentiment Analysis, Text Summarization, Machine Translation, and Answer Relations. This is because CNN has the capability of dealing with data parallelly. CNN among its other variants like RNN is preferred on account of fast response and possible implementation on GPU units.

The receptive field for text processing convolution neural networks is created by striding filter over words instead of image patches. We feed words as matrices into the input layer of the neural networks. Variation in the size of the Kernel may be helpful in detecting different kinds of patterns in text. The pattern is an n-gram that can be located anywhere in the document regardless of its position.

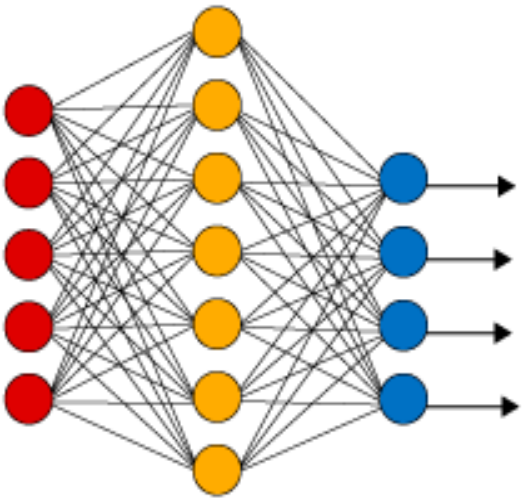

Figure 2.10: Architecture of Convolution Neural Network

Figure 2.10 shows how CNN performs classification on documents. A matrix embedding for the sentence is fed to it and the different sizes of filters are convoluted to evaluate features. It results in 6 features that are concatenated to generate a feature vector for sentence representation. The softmax layer receives this vector as input and classifies the document.

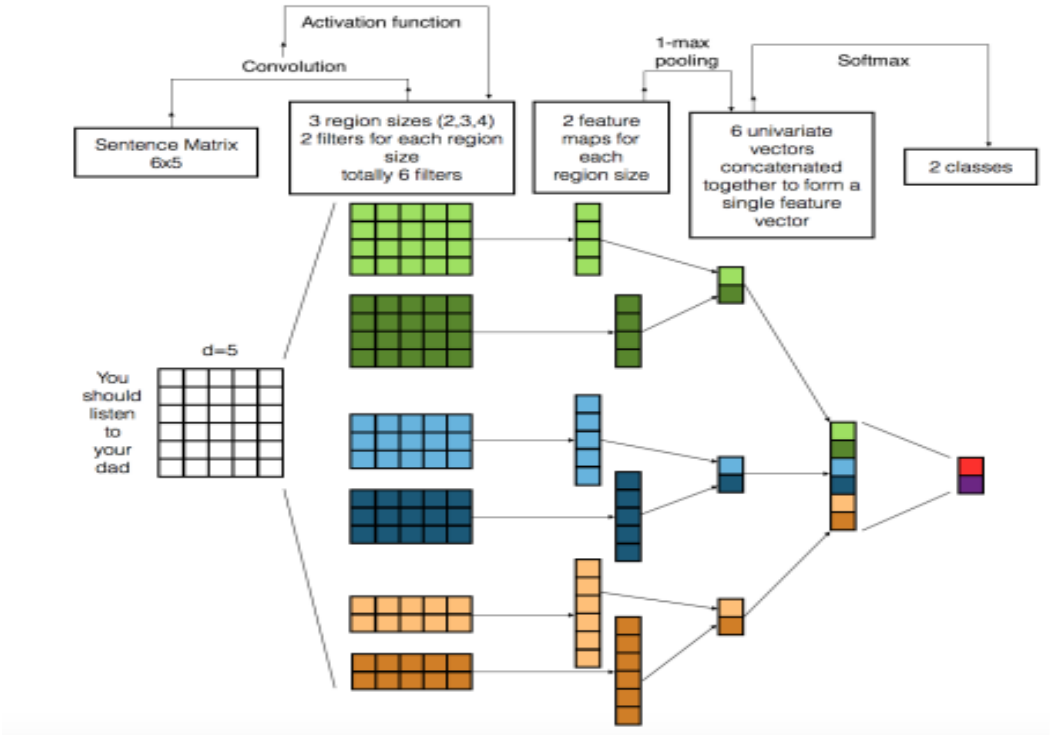

Figure 2.11: Document Classification using CNN Moreno et al (2017) [35]

Pooling plays an important role in convolution neural networks. It helps to reduce spatial size and number of the parameters by down sampling the convolution results which subsequently avoids over-fitting. Some of the common pooling methods are max-pooling and avg-pooling. The pooling procedure gives another form of translation in-variance. Pooling can be performed over a matrix or window. In order to get a fix size matrix pooling is evaluated at the end of the result in natural language processing [35].

In other words, Pooling reduces output dimensionality but retains the most relevant information. By performing the max procedure, only the most significant features are sampled: as a result, global localization information (where a pattern was identified in a sentence) is lost, but local information captured by the filters (features generated by high-value convolutions) is retained.

Rectified linear units (ReLU) are placed after the output of every layer to ensure non-linear properties and omit negative results from matrices. It trains network faster without introducing any penalty, in our model we have utilized tanh activation function. Parts of Speech tagging and entity extraction are considered unfit for CNN because of the loss of information at pooling layers.

Huges (2017) [36] used CNN to classify text from two medical datasets. He evaluated the model with various filter sizes and configurations. The best performing network was obtained with a pair of two convolution layers followed by SoftMax. Later drop-out of 0.5 was introduced to avoid overfitting, the model was tested against state-of-the-art BOW and Doc2Vec model and it outperformed in various evaluations.

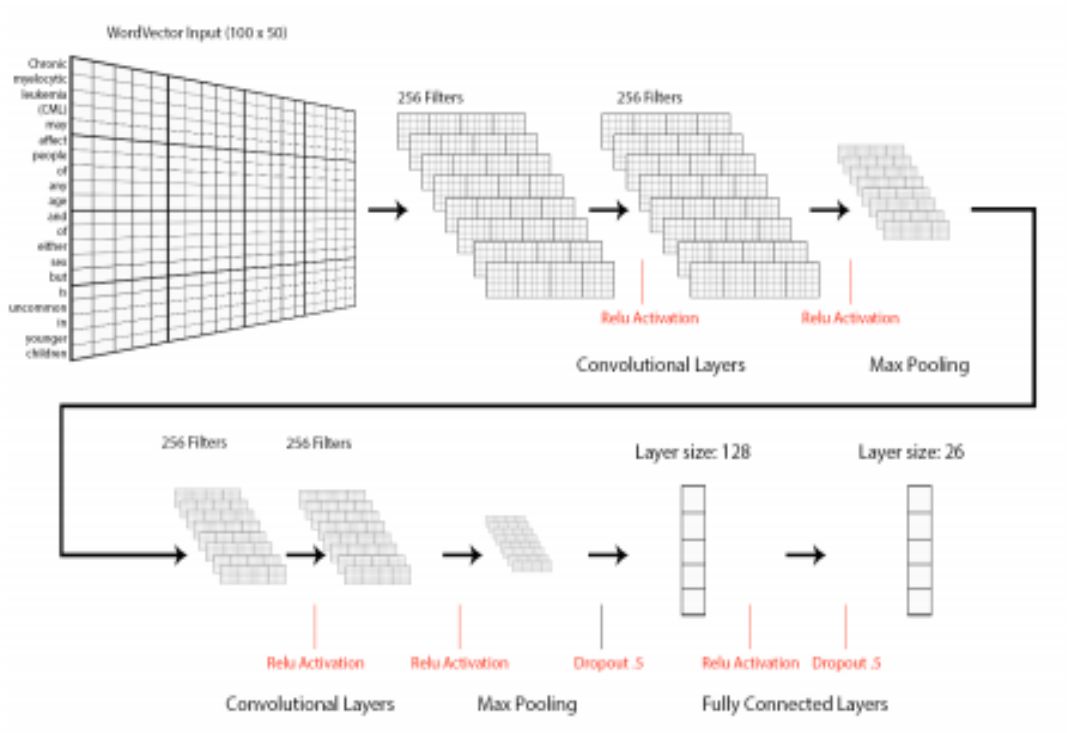

Figure 2.12: CNN Model by Hughes (2017) [36]

2.4.1 Dilated Convolutions

Fisher Yu et at (2015) [37] presented the idea of dilation in convolution neural networks from wavelet decomposition. In particular, dilation refers to the size of spatial coverage by Kernel. Variation in the size of the kernel allows covering a wider range from the input. In NLP, it will naturally help to preserve semantic meaning from longer sentences be creating an n-gram. Considering * as discrete convolution operator and l as dilation factor, F as a function and K as kernel size then we can describe it as :

$$(F * K)(p) = \sum_{k+lt=p} F(s) * F(t)$$

In the equation placing l = 1 changes it to standard convolution. The dilated convolution operator can use the same filter at various ranges using different dilation factors. The definition reflects the proper implementation of the dilated convolution operator, which does not involve the construction of dilated filters.

Figure 2.13 shows the variation in the dilation layer covers a longer area of the receptive field. A similar trend can be observed in the text. In our model, we have used five dilation layers with 1,2,4,8,16 kernels. Which helps to create 64-gram which is enough to cover the longest sentence in discharge summaries. Figure 2.14 shows the rough estimation of flow from dilation to the text.

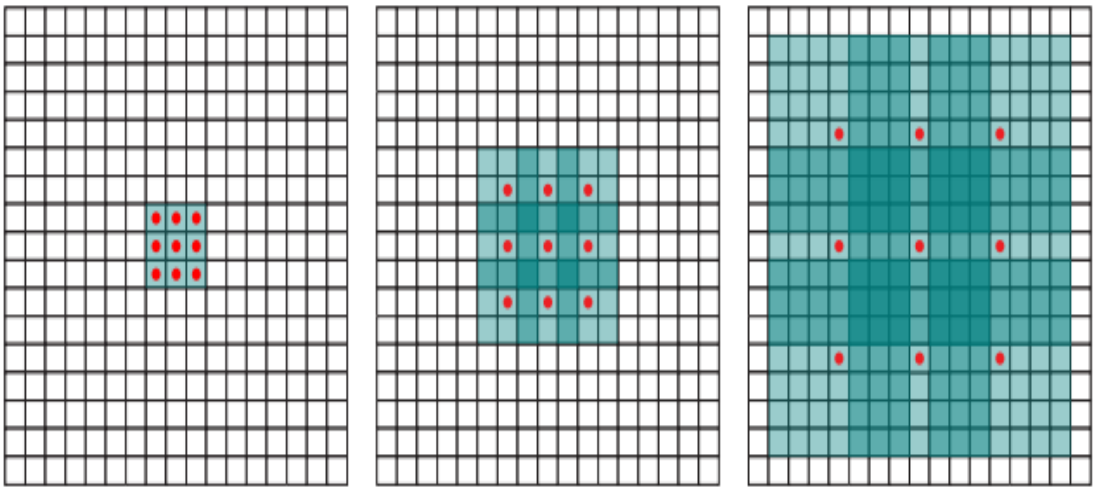

Figure 2.13: Dilation at 3x3,7x7,15x15 spatial field [37]

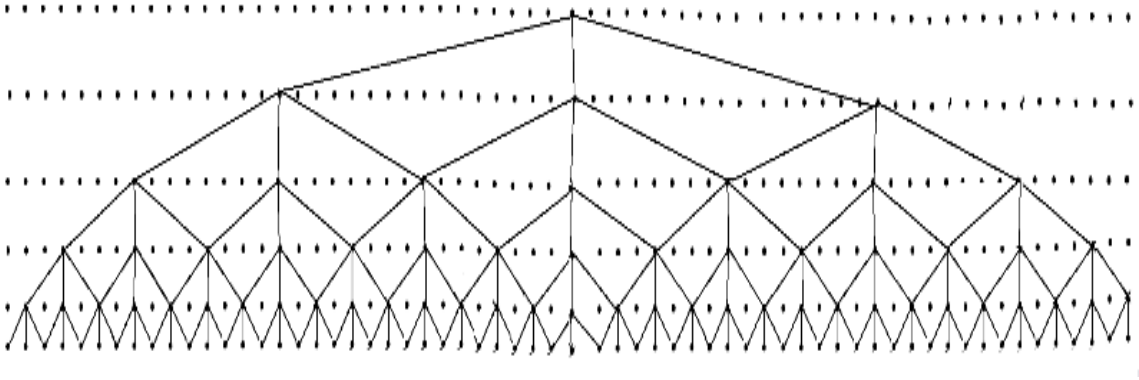

Figure 2.14: Dilation forming 64-gram for 5 layers

2.5 ICD Health-Care System

World Health Organization (WHO) among its many other initiatives maintains and develops the International Classification of Diseases (ICD) Health management system. This system has standardized a hierarchy of codes for all kinds of medical diseases, procedures, and surgical operations. It categorizes a wide variety of symptoms with its child codes for every main code. Codes may be six-digit longer based on specific variations. First, worldwide procedural codes were published in 1949, namely ICD-6. Eventual development in medical science made it necessary to create frequent updates and in 1978 final ICD-9 was presented [49].

ICD-10 is currently used and is available multi-lingually. It contains nearly 155,000 codes for new procedures and diagnoses. It has two variant ICD10-AM and ICD10-CA based on country guidelines. ICD-11 is supposed to become from January 2022. It will contain the addition of Diagnostic and Static Mental Disorders (DMS). It is supposed to be easier and semantically reliable than all previous versions [49].

RANGE	Class of Disease/Symptom
001-139	Infectious and parasitic diseases
140-239	Cancer
240-279	endocrine glands, nutrition , metabolism, and disorders immune
280-289	blood and hematopoietic organs
290-319	Mental Disorders
320-389	nervous system and sense organs
390-459	circulatory system
460-519	respiratory system
520-579	digestive system
580-629	genitourinary system
630-677	pregnancy, childbirth, and the puerperium
680-709	Skin and tissue diseases
710-739	musculoskeletal system
740-759	Congenital malformations
760-779	conditions of perinatal
780-799	Symptoms and signs of morbid states
800-999	Traumatism and poisoning

Table 2.2: Code Ranges in ICD-9-CM

Most of the ICD-9-CM codes are numeric, while the additional classifications contain alphanumeric codes. All are composed of three, four or five characters and each has a specific description. The classification contains over 12,400 final diagnosis codes and about 3,700 final procedure codes. Final ICD-9-CM integrates all the official updates published from October 1986 to October 2006. This classification only contains numeric codes, between 001 and 999.9 [12].

The sections between 01 and 86 include major surgery, endoscopies, and biopsies. Headings 87 through 99 include other diagnostic and therapeutic procedures and are grouped on the basis of the type of procedure. In table 2.2 below is described details of code ranges according to specific diseases.

Given ICD-9-Clinical Modification, the structure is aggregated further into subcodes. Where the main code head (parent code) refers to the generic concept and lower tail (Child Code) shows a specific procedure or condition. Additional information on the organization can be obtained from BioPortal where all root level details are described. The average number of children to code head is five but some codes have a maximum of 21 children. This hierarchy is imbalanced and grouped based on codes relating to similar nature like all procedures are lined in a row.

MIMIC (Multiparameter intelligent monitoring intensive care) Dataset was labeled following guidelines dictated by Unites States National Center for Health Statistics (NCHS). All clinical notes from MIMIC are marked with single or multiple codes of ICD-9 CM. ICD9-CM is the additive abstraction of ICD-9. There are three volumes of this coding system. The first volume is a tabular index with a list of disease codes. Second is an alphabetical index to disease entries and the last volume is for therapeutics procedures with alphanumeric codes.

Code	Labeled Disease
16	Tuberculosis of genitourinary system
16.0	Tuberculosis of kidney
16.00	Tuberculosis of kidney, unspecified examination
16.01	Tuberculosis of kidney, bacteriological or histological examination not done
16.02	Tuberculosis of kidney, bacteriological or histological examination unknown (at present)
16.03	Tuberculosis of kidney, tubercle bacilli found (in sputum) by microscopy
16.04	Tuberculosis of kidney, tubercle bacilli not found (in sputum) by microscopy, but found by bacterial culture
16.05	Tuberculosis of kidney, tubercle bacilli not found by bacteriological examination, but tuberculosis confirmed
16.06	Tuberculosis of kidney, tubercle bacilli not found by bacteriological or histological examination.

Table 2.3: A layout from ICD-9 Hierarchy

In order to issue a patient's codes, a labeler must first decide the cause for the patient's appointment by analysing the signs, symptoms, diagnoses, and impediments recorded in the doctor's notes. Conditions classified as "possible" must not be labelled and only specific signs and diseases must be considered. Each medical condition is classifiable only in one of the final groupings of the classification [45]. Some most used codes in MIMIC-III are shown in table 2.4.

Codes	Type of Disease	Assignments
427.31	Atrial fibrillation	17903
401.9	Unspecified essential hypertension	17903
428	Congestive heart failure, unspecified	11731
584.9	Acute kidney failure	7926
518.81	Acute respiratory failure	6490

Table 2.4: 5 Most Frequent codes in MIMIC-III

During the last decade, Artificial Intelligence research has been expedited especially in sectors where text-processing has a significant role. It is difficult to equally compare different methods because they are formed on different points of Dataset. In this section, an overview from existing work (Lita et al. 2008; Perotte et al,2013; Kavuluru et al 2015; Yang et al, 2016; Shi et al, 2017; Crammer et al,2017; Baumel et al, 2017; Mullenbach et al 2018) will be presented.

2.6 Background

In the study, conducted by Lita et al (2008) [30]. Two models, Support Vector Machines (SVM) and Bayesian Ridge Regression were tested to understand word distribution in clinical text. In SVM, a positive to negative example ratio from the targeted test was applied with a linear kernel. Gaussian process in a probabilistic approach sought in BRR. Ridge Regression computes the likelihood function from labels conditioned to weighted inputs of the document. Both Support Vector Machines and Bayesian ridge regression methods are fast to train and achieve comparable results. Evaluation is based on the precision

score on top 5 codes with balanced distribution. BRR outperform SVM by 3% standing at 65.7% while F-1 Score show contrasting (78.4% vs 77.2 %) for SVM and BRR respectively [30].

Perotte et al. (2013) experimented with flat and hierarchic SVMs with MIMIC-II data collection TFIDF functionality, using ICD9 hierarchy and only predicts child codes with positive parent codes. The hierarchical SVM also considers the structure of the ICD9 code tree during training. An increased data set is created, where each document is labeled with codes from the whole tree, not just the leaf nodes. During training many SVM classifiers are created and trained, one for each code of the hierarchy excluding the root [16].

The classifier associated with the code in the hierarchy is used only if its parent code has been classified as positive. Therefore, only documents with a positive parent code are fed to the child classifier. Hierarchy-based classification produces better ICD9 coding results than flat classification for MIMIC patients. The classifier works from the root down to classify until negative codes are found. The procedure is repeated to result in a tree for multi-label classification for a given document.

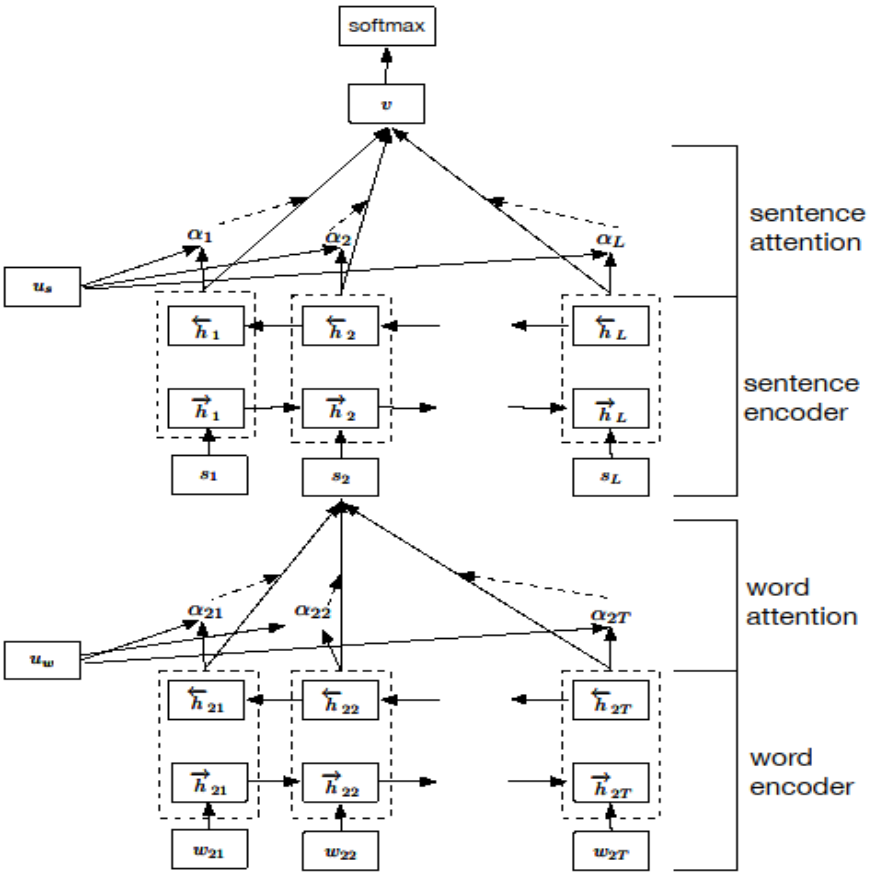

Figure 2.15: Hierarchal Attention from Yang et al

The flat SVM views every label as an individual binary decision. For each possible ICD9 code, one linear distinct SVM classifier is generated, except for the root, which is always positive. Those documents in the training set classified with the ICD9 code are considered to be valid and all other documents are considered to be negative. Positive ICD9 code's head code must also be positive, and negative ICD9

code descendants must also be negative. Only during the test setups, when single predictions include all ancestors, is this relationship taken into consideration.

The outcomes of the study show hierarchy based SVM performs better than flat SVM in some evaluation criteria results in improved recall (30% vs 16.4%) with the expense of precision (57.7% vs 86.7%). However, he hierarchy-based SVM reached 70.0% recall and 83.6% precision for cerebral artery occlusions [16].

Ramakanth Kavuluru et al (2015) built a classifier on ICD-9-CM labeled data from the Medical Center of the University of Kentucky. Their proposed model showed a learning-to-rank based binary approach for different scales of the dataset. The problem was approached with logistic regression, SVM and naïve base feature selection [20].

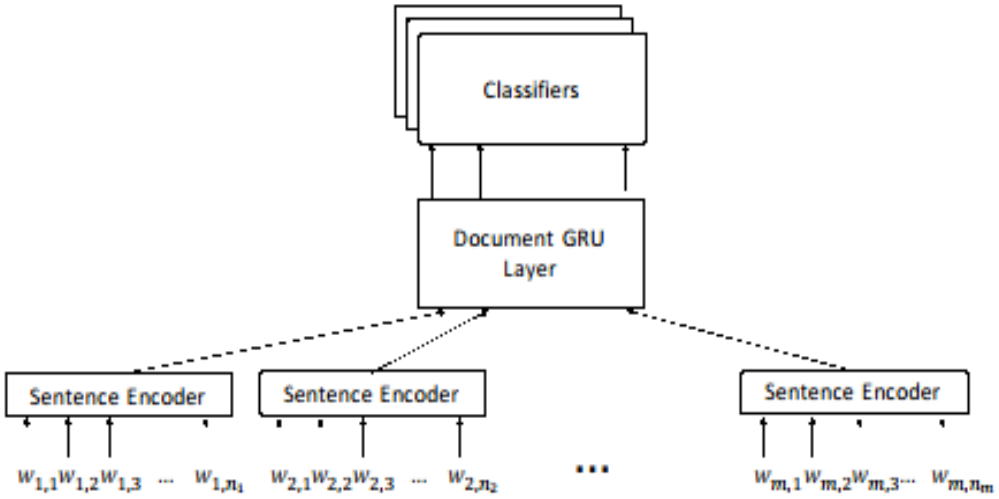

Figure 2.16: Hierarchal GRU Model (Baumel et al)

In all codes, they achieved a micro F-score of 0.48 with at least 50 training cases. For regard to the collection of codes that appear at least 1 percent in the two years data set, a micro F-score of 0.54 was achieved [20].

Yang et al (2016) proposed a hierarchal attention network with two distinct characteristics. Primarily, a hierarchal architecture for structural documents and distributed attention for word and sentence level. The model comprises of numerous parts: a sentence encoder, a sentence-level attention layer, word sequence encoder, and a word-level attention layer. The model suggested projections of the raw text into a vector representation on which we construct a classifier for the classification of documents. Figure 2.15 above is the structure of flow in the model. Results show evaluation on Yelp and Yahoo answers. Hierarchal Attentional networks with Gated Recurrent Unit achieved 0.682 and 0.758 precision respectively [18].

Baumel et al (2017) experimented on MIMIC-II and MIMIC-III with four different models. He observed code assignment to discharge summaries with SVM-based one-vs-all model, a bidirectional Gated Recurrent Unit (HA-GRU), a convolutional neural network (CNN) model and a continuous bag-of-words

(CBOW) model. Baumel applied hierarchical segmentation for pre-processing. He proposed two motivations for preprocessing the input texts with sentence breakdown. First, it is impractical and ineffective to train a sequence model like a GRU on such long sequences. Secondly, he chose to implement a moving window for discharge summaries [11].

The continuous BOW model was an applicative use of Mikolov et al (2013) model [15]. This was based on creating embedding of fix-size. Instead of averaging embedding, a One-dimensional convolutional filter is applied. Hierarchal (HA-GRU) is set to handle multi-label classification where instead of applying a common GRU on the whole document, which is a certainly slower approach. Two-level hierarchal bidirectional GRU encoding is used. The second GRU unit was aimed at keeping attention on each label. The input text sentence will be encoded to a fixed-length (64) vector by using an embedding layer on all inputs, applying a bidirectional GRU layer on the embedded words and encoding bidirectional GRU outputs using a neural attention mechanism (size 128).

The Result indicates that the HA-GRU model outperforms CNN, CBOW, and SVM with the micro-F measure. The CBOW model score 43.30% compared to 55.86% of HA-GRU. Collectively, Hierarchal Attentional GRU 7.4% and 3.2% improvements over CNN and SVM. The ability to feature the decision process of the model is important for the implementation of such models by medical professionals. Conclusively, HA-GRU needs less training data to achieve top performance, which is crucial for domain adaptation efforts when applying such models to patient records from other sources.

Crammer et al (2017) formed a learning system based on features. In every document input and output were built with a vector-valued representation where each input corresponds to one or few output variables. A linear model of weight vectors is used to rank the labels. Specific Features were indexed with their names and presented with specific labels. MIRA algorithm is used to create weight vectors during learning for each input document. A rule-based approach was set up to deal with uncommon configuration. The recall score at the top 30 codes approached 76.6% compared to their baseline score of 72.58% [13].

Horan Shi et al (2017) [19] offered attention-based character aware neural language models to classify diagnosis descriptions. The model has prepared on a portion of the top 50 codes from MIMC-III summaries. Which included 8,066 hospital admissions. The coding model consists of four modules. A diagnosis description encoder, ICD Code encoder, attention for assigning codes and matching text with relevant code. The detailed model is shown in figure 2.16 .

Encoding is leveraged on a Long-Short term memory LSTM network which a variant of recurrent neural networks (RNN). LSTM is efficient here because of a long diagnosis text document. Hidden representation for each input is obtained with character level LSTM and Word Level LSTM. Character level encoding is expected to catch better features because of their many medical terms with the same suffix. The input is the sequence of sentence matrices in the word-level LSTM, each obtained by concatenating the word representations derived from the previous layer. The last hidden state is the interpretation of each sentence. The same two-level LSTM architecture is implemented for embedding descriptions of ICD codes in order to obtain the hidden representations. The layer parameters for the code definition encoder and the document encoder are separate in order to better respond to the different language forms in each of the two text sets [19].

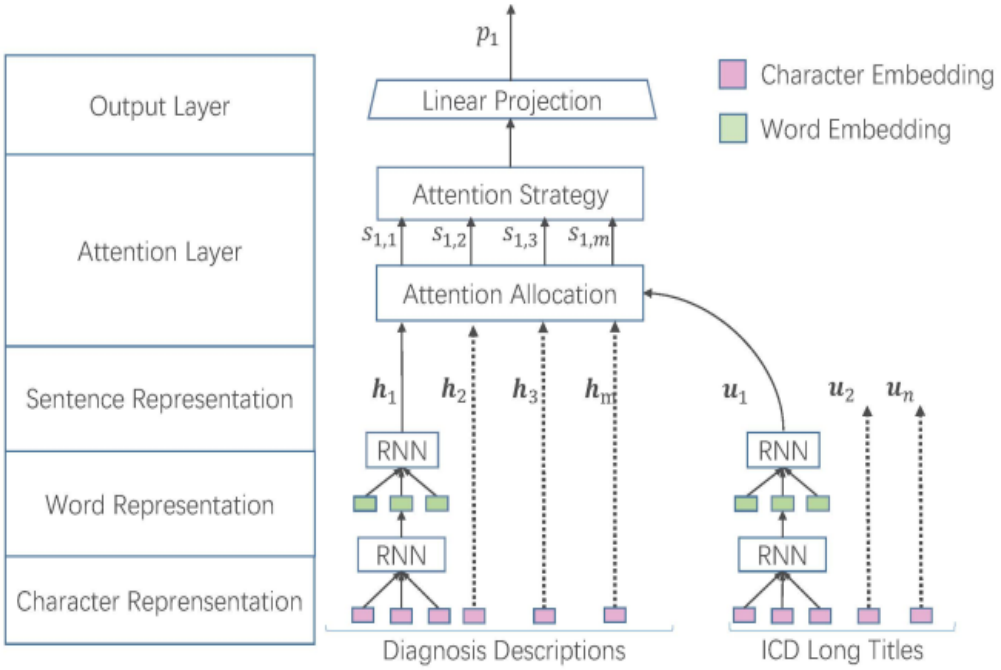

Figure 2.17: Model From Original Paper (Shi et al 2017) [19]

The number of written diagnosis summary is not equal to the number of ICD code assigned, because one code for the same diagnostics description cannot explicitly be allocated. The attention mechanism offers a recipe to determine the diagnostic details for coding are relevant. It evaluates cosine similarity between text and ICD code before supplying it to the SoftMax classifier. Finally, we use sigmoid to normalize the confidence score into a probability value. F1 and area under the curve (AUC) scores for soft attention models are 0.532 and 0.900 correspondingly.

James Mullenbach et al (2018) [8], in their work, "Explainable Prediction of Medical Codes from Clinical Text" proposed a variant of Yang et al (2016) work on hierarchical attention network with Convolution neural networks. The proposed method Convolutional Attention for Multi-Label classification (CAML) was based on a per-label attention mechanism. Instead of taking whole representation after aggregating their proposed applying attention to the area of the document where the relevant text is present. Moving weighted matrix to output layer applying a sigmoid function to find the likelihood of code. Pre-trained embeddings are formed for each token in the document in the form of a one-dimensional vector. An element-wise non-linear transformation is applied with a fix size filter and unitary stride.

Input is padded to form the output matrices of similar dimensions. The result of the operation is moved to computer attention, the target is designed to create multiple labels using k-grams. A later attention vector with lower dimensions is used to compute vector representation for each label. Training of the model was aimed at reducing binary-cross entropy and penalizing L2 weights with Adam Optimizer.

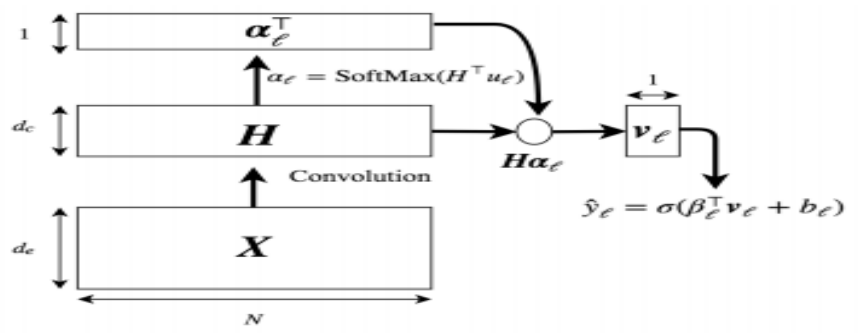

Figure 2.18: CAML Architecture (Mullenbach et al 2018) [8]

The model is validated broadly for MIMIC-II, MIMIC- III, and top k codes. A recent work has provided vast evaluations in order to compare the model with Baumel et al (2017) [11] and Crammer et al (2017) [13]. He presented precision scores for most common eight and fifteen labels across the dataset. Additionally, micro and macro scores have shown significant improvement in their baseline models. Their best model achieved 0.709 scores for precision on top 8 codes and 0.539 for micro F-1 measure [13].

Analysing the relevant studies we can say The CAML architecture suggested in this paper shows better results under all experimental circumstances. Recurrent neural networks perform slower than convolution neural networks because of their sequential flow. Moreover, longer text documents do not essentially preserve the semantic understanding in LSTM.

Improvements are due to the mechanism of attention which focuses on the most critical elements for each code instead of using a standardized pooling procedure for every code. We also found that convolution-based models are at least as effective as recurring neural networks such as the Bi-GRU, and substantially more computationally powerful. Models that exploit the hierarchy of ICD codes and attempt the more difficult task of predicting diagnosis and treatment codes for future visits from discharge summaries are better from the application perspective.

CHAPTER 3

Methodology

Despite this progress, the digital system's interoperability continues to be an open problem that poses challenges in data integration and processing for a model. The capacity for the interpretation and enhancement of treatment offered by hospital data is therefore still to be fully realized. At the same moment, there are increasing concerns from the scientific community that experiments are not quantifiable. Based on previous MIMIC (MIMIC-II), which was published in 2010 MIMIC-III, we predict that MIMIC-III will be commonly used worldwide in fields such as scientific and industrial science, quality improvement programs and higher education.

The Computational Physiology Laboratory is an interdisciplinary unit of data scientists and clinicians at the Massachusetts Institute of Technology. MIMIC-III is the third MIMIC critical care database version and helps us to build on previous data processing and development expertise. The MIMIC-III database was filled with data collected during routine hospital care, so patients were not burdened, and their workflow was not interrupted. The information has been downloaded from various sources, including the Social Security Administration's death master file, electronic health record databases, critical care information system documents [45].

3.1 Information Flow

For the prediction of ICD9 codes from clinical notes, we are using the table of NOTEEVENTS which are free text notes providing progress notes and hospital discharge summaries. Many previous studies like Baumel et al (2017) and Mullenbach et al (2018) used discharge summaries or discharge procedures[11][8]. Only a few others like Kavuluru et al (2015) used patient stays from other databases of UKY [20]. For each patient entry in discharge summary ICD-9 code is present (Labeled 1) or (Labeled 0) and likelihood ranges between 0-1. Output values that are higher than 0.5 are counted for the presence of code.

In this work, we are using diagnosis and procedure codes to combine for localizing patient summaries. We are using a full dataset with 9017 codes. Processing of clinical notes with five-layer dilated convolution networks gives better results than all previous Mullenbach et al (2018) [8] and Shi et al (2017)

[19]. We initialized embedding from pre-trained distributed models. Training contains the choice to either evaluated embedding using Word2Vec or GloVE 42B with 300 dimensions. There is an additional option to create stack embedding because many relevant embeddings are not able to any of these models. So, we create random embeddings for out-of-vocabulary terms. Stack embeddings are supposed to increase coverage external embeddings from glove are Common English corpus instead of any specialized medical corpus. In order to protect the health information and input tokens, these embeddings are kept trainable to divert according to the relevant context [19] [8].

We are experimenting with the model used by Stefano et al (2019) [43] with a variant of computing embeddings with word level. The model computes dilation at the sentence level by changing filter size from lower to a higher level to compute attention in a hierarchal way similar to Yang et al (2016) [18]. Model formed by Mullenbech et al (2018) was tuned for top 50 codes and parameters were optimized for top 8 occurring codes whereas we propose a model for full codes in MIMIC-III. Attention is scored based on embedding descriptions at the word level.

3.2 Pre-processing

The creation of the MIMIC dataset involved balancing interpretation simplicity against proximity to ground-based facts. As such, information is a representation of the underlying data sources, updated in response to user feedback through the MIMIC database iterations. The problem with the underlying text is its uncommon nature. As discussed previously in section 2.5 ICD Coding system, the medical text is different in terms of acronyms, short form. Medical drug names, vital signs and abbreviations. Applying simple lemmatization and stemming tools may lose the semantic meaning of diagnosis notes.

Additionally, these notes are de-identified based on regulations of (H.I.P.P.A) to protect health information. Information like Lastname, first name, location, dates is removed and is replaced with generic unknown tokens. For Instance [** Known Lastname 025046**], the purpose of putting this number is to keep human identification hidden from the researcher. MIMIC presents subject_ID to interpret the related history to every person. Various text-preprocessing approached were tried on MIMIC, Baumel et al (2017) replaced non-alphabetical characters to pseudo tokens [11].

Shi et al (2017) [19] transformed the messy and inconsistent raw note texts in these parts into tidy diagnosis details. He used a number of regular text pre-processing techniques such as regular expression matching and tokenization. That resultant mark is a short sentence or a term that articulates a disorder or illness. In order to evaluate the model in a common way, he visited patients that do not include details of the condition are rejected and worked on top 50 codes.

The strategy of removing tokens that do not contain alphabetical characters (e.g. remove "500" but keep "250 mg"), lower all tokens, and replace tokens that appear with a ’ UNK ’ token in less than three training documents[8]. He cropped documents at a maximum of 2500 token in each summary.

The text processing data consumed by this work is similar to the work in which we are using SpaCy to

create custom tokenization with the rule-based approach [43]. The prime reason behind this approach is the nature of MIMIC notes described in section 4.1. To treat the most common medical abbreviations accurately, as the tokenizer sometimes breaks them incorrectly. Therefore we delete tokens with no alphabetic characters: this law excludes numbers and bogus tokens made entirely of symbols, but it preserves words containing numbers like (from 60 Capsules, removing 60).

This mapping retains the text structure, is easily interpretable when interpreting the written annotations, and allows such words to be accurately tokenized. We have checked the numerical attributes associated with the anonymized tokens: just as [**Unknown 5252**] is set to be preserved as "unknown_5252". We found that this strategy resulted in a significant increase in the number of distinct tokens in the body, leading to yet more complex outcomes. Also, dates are ignored just the name of months is taken as a string from discharge and admission dates for example 2008/8/24 is left as "august".

Similar to Tal Baumel et al (2017) [11] approach of dealing with out-of-vocabulary words, We took all those tokens that repeated at least three times in training corpus. Special UNK token is introduced for OOV words. The approach of shortest Levenshtein distance was implemented to map unknown words to known words but it does not show any significant development in evaluation metrics [14].

Here is an example that illustrates some of these tokenization excerpts: anonymized dates and names, organized alphabetically lists, information on drug delivery, essential signs, etc. This illustrates a sample of clinical notes before or after our preprocessing and tokenization. The note processed with a one-sentence-per-line format is generally clearer and we can observe how most problem expressions are treated correctly. The main problem is the detection of the sentence boundaries, with many phrases incorrectly separated into two or more phrases.

3.3 Model Overview

The model is based on three components, embedding layer, convolution layers, and attention layer. Data is fed to embedding layers which then initializes matrices for text present and forwards to the convolution layers. These embeddings can be computed in various ways with selective option to choose Word2Vec, Fasttext English common crawl of GloVe with forty-two billion tokens. Although this corpus is not medical specific, it yields a good coverage for text present in discharge summaries. Dilation layers create an overview of sentence-level understanding to create attention at sentence vectors.

Document D comprises of L sentences s_i, where i presents the number of the sentence. Assuming each sentence holds T_i words where w_{it},t ∈ [1,L] and t ∈ [1,T_i] represents T^{th} word in the ith sentence. Each word is wit mapped to a unique embedding vector through the embedding word matrix W, we are using embeddings of 300-dimension. Where this can be mathematically described as

$$X_{\text{it}} = W * w_{\text{it}}$$

Similar to the work [43], The words that form a description are mapped through the same embedding matrix W used for the documents to their continuous embedding space. The resulting sequence of vectors is then applied to a non-linear projection. A max-pooling operation reduces the sequence to a vector, followed by a final non-linear projection. The representation of each code that has been obtained is then used as a query vector to calculate the attention scores for that code. For every code c, corresponding context vector uc is computed from its description dc using:

$$W_{\mathrm{c}} = W * d_{\mathrm{c}}$$

$$\widetilde{(W_c)} = \tanh(W_{a1} w_c + b_c)$$

$$h_c = maxpool(\widetilde{(W_c)})$$

$$u_c = \tanh(W_{a1} h_c + b_{a2})$$

Attention has been described previously in section 2.3, It is really important to infer the key information from text. For words-level attention representation, word representation hit goes through a fully connected layer and a non-linear activation function.

$$u_{it} = \tanh(W_w h_{it} + b_w)$$

The importance of each word can be estimated through its similarity with a uw vector normalized using SoftMax. This normalization is obtained using exponential nonlinearity.

$$\alpha_{it} = \frac{exp(u_{it} T u_W)}{\sum_T exp(u_{it} T u_W)}$$

To obtain a final vector representation of the input at sentence level s_i, the sum of the input vectors measured by their respective attention values is determined.

$$s_i = \sum_t^T \alpha_{it} h$$

As a function of the input attention is formally measured as a set of key-value pairs K, V and Q query matrix. Next, a function f measures the similarities between each Ki key and the query to achieve weight. A common similarity function used is the dot product, but it is also possible to use other functions. In order to normalize these weights, a SoftMax function is then used. Finally, the weights of focus are used to measure a weighted sum of the values and to achieve the final representation.

$$F(Q, K_i) = Q_T * K_i$$

$$a_i = softmax(F(Q, K_i))$$

$$Attention(Q, K, V) = \sum_i a_i V_i$$

In our model K=V=h refers to the output of convolutions. For each label, c attention is a product of h and u_c where u_c is query vector. Every element of a_i vector includes the sequence V's corresponding element's attention value. The sum of the V_i components measured by a_i focus scores creates a vector text V.

High-level representation is held by the document vector v which can be used for classification tasks. This document representation is fed to a fully connected layer and passed by sigmoid function β to compute the final likelihood of each label c.

$$y_c = \beta(W_c\ V + b_c)$$

During the training, the classifier tries to fit the model to training points. We are using binary cross-entropy to compute loss on the output layer. The model is trained to minimize this loss.

$$LBCE\left(y, \widetilde{y}\right) = -\sum_{c=1}^{C} y_c \log\left(\widetilde{y_c}\right) + (1 - y_c)\log(1 - \widetilde{y_c})$$

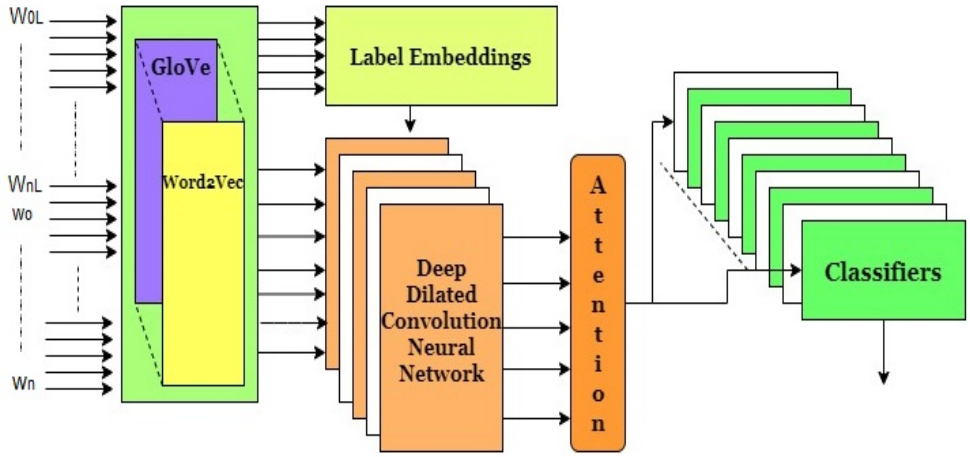

Figure 3.3: Model Flow Diagram

Admission Date: [**2118-6-2**] Discharge Date: [**2118-6-14**]

HISTORY OF PRESENT ILLNESS: This is an 81-year-old female with a history of emphysema (not on home O2), who presents with three days of shortness of breath thought by her primary care doctor to be a COPD flare. Two days prior to admission, she was started on a prednisone taper and one day prior to admission she required oxygen at home in order to maintain oxygen saturation greater than 90%. She has also been on levofloxacin and nebulizers, and was not getting better, and presented to the [**Hospital1 18**] Emergency Room. In the [**Hospital3 **] Emergency Room, her oxygen saturation was 100% on CPAP. She was not able to be weaned off of this despite nebulizer treatment and Solu-Medrol 125 mg IV x2.

Review of systems is negative for the following:

Fevers, chills, nausea, vomiting, night sweats, change in weight, gastrointestinal complaints, neurologic changes, rashes, palpitations, orthopnea. Is positive for the following: Chest pressure occasionally with shortness of breath with exertion, some shortness of breath that is positionally related, but is improved with nebulizer treatment.

PAST MEDICAL HISTORY:

COPD. Last pulmonary function tests in [**2117-11-3**] demonstrated a FVC of 52% of predicted, a FEV1 of 54% of predicted, a MMF of 23% of predicted, and a FEV1:FVC ratio of 67% of predicted, that does not improve with bronchodilator treatment. The FVC, however, does significantly improve with bronchodilator treatment consistent with her known reversible air flow obstruction in addition to an underlying restrictive ventilatory defect. The patient has never been on home oxygen prior to this recent episode. She has never been on steroid taper or been intubated in the past.

DISCHARGE MEDICATIONS:

1. Levothyroxine 75 mcg p.o. q.d.

2. Citalopram 10 mg p.o. q.d.

Figure 3.1: A section from MIMIC Discharge Summary

admission date june discharge date june

history of present illness this is an year old female with a history of emphysema not on home o2 who presents with three days of shortness of breath thought by her primary care doctor to be a copd flare two days prior to admission she was started on a prednisone taper and one day prior to admission she required oxygen at home in order to maintain oxygen saturation greater than she has also been on levofloxacin and nebulizers and was not getting better and presented to the hospitaltoken emergency room in the hospitaltoken emergency room her oxygen saturation was on cpap she was not able to be weaned off of this despite nebulizer treatment and solu medrol mg iv x2

review of systems is negative for the following fevers chills nausea vomiting night sweats change in weight gastrointestinal complaints neurologic changes rashes palpitations orthopnea is positive for the following chest pressure occasionally with shortness of breath with exertion some shortness of breath that is positionally related but is improved with nebulizer treatment

past medical history copd last pulmonary function tests in november demonstrated a fvc of of predicted a fev1 of of predicted a mmf of of predicted and a fev1 fvc ratio of of predicted that does not improve with bronchodilator treatment the fvc however does significantly improve with bronchodilator treatment consistent with her known reversible air flow obstruction in addition to an underlying restrictive ventilatory defect the patient has never been on home oxygen prior to this recent episode she has never been on steroid taper or been intubated in the past

discharge medications

levothyroxine mcg p.o. q.d.

citalopram mg p.o. q.d.

aspirin mg p.o. q.d.

Figure 3.2: Preprocessed discharge summary

CHAPTER 4

Experimental Setup

A common issue with classification of clinical notes is constructing interference between model loss and assigned labels. This mapping generalizes models for code prediction to sensible clinical data. Nevertheless, it was seldom used for automatic assigning to medical records with medical codes such as ICD9. Much of it is because it is difficult to obtain data and labels. For instance, the online versions of `http:/www.icd9coding.com/` are mostly used by ICD9 code assignment systems with a rule-based engine.

Hospitals are typically hesitant to share their patient data with research groups and sensitive information (e.g. patient name, date of birth, home address, social security number) must be anonymized to comply with HIPAA requirements. Protected health information was excluded from free text areas, such as diagnostic records and medical notes, using a rigorously validated de-identification system based on extensive dictionary searches and regular expression patterns. When new data are obtained, the components of this monitoring system are continuously expanded.

4.1 The MIMIC Dataset

MIMIC (Medical Information Mart for Intensive Care) is an open-source clinical database. It contains information on patients confined to critical care facilities in a major tertiary hospital. The data includes critical information, medication, laboratory tests, care provider findings and notes, fluid balance, procedure codes, diagnostical codes, photo reports, stay-length hospital data and more. Recovery data. The database serves projects such as scientific and market research, programs to improve quality and higher education [45].

The first release is known as MIMIC- II (Multiparameter Intelligent Monitoring Intensive Care) which contained the data of patients at Beth Israel Deaconess Medical Center between 2001 and 2008. Since the MIMIC was one of the few first available databases, A lot of research publication is concluded based on MIMIC-II. To relate it with the current data website `https://mimic.physionet.org/mimicdata/whatsnew/` describes the relationship between tables to understand how it was upgraded. MIMIC-III is an extension of old MIMIC-II which was later incorporated with further data from 2008–12. This transition was done in several queries some items like D_MEDITEMS, D_IOITEMS, D_CHARTITEMS

were merged to D_ITEMS. Admissions and Discharges were labeled with a time component. Moreover, CENSUSEVENTS replaced by TRANSFERS, DEMOGRAPHIC_DETAIL merged into ADMISSIONS DRGEVENTS renamed DRGCODES, ICD9 renamed DIAGNOSES_ICD and so on [45].

Currently, in this work, we have used version is MIMIC-III (v1.4), released in 2016. MIMIC-III includes data associated with 53,432 adults and 8100 new-born children admitted to critical care units between 2001 and 2012. In addition. The median age of adult patients is 65.8 years. The median length of an ICU stay is 2.1 days and the median length of a hospital stay is 6.9 days. A mean of 4579 charted observations ('chart events') and 380 laboratory measurements ('labevents') are available for each hospital admission. Table 4.1 provides a breakdown of the adult population by the care unit.

MIMIC-III is presented as a compilation of comma-separated value (CSV) files. A free demo version of 100 records can be downloaded directly from Physionet but in order to access all 26 tables, a special MIT based course is required. The course "Data or Specimens Only Research" course ensures defining data regulation laws for research purposes. Here is a table presenting details of data. The tables are associated with identifiers that usually have the suffix 'ID'. For example, SUBJECT_ID describes to a unique patient, HADM_ID points to a unique admission to the hospital, and ICUSTAY_ID refers to a single admission to an intensive care unit.

Table Name	**Description**
ADMISSIONS	Every unique hospitalization for each patient in the database (defines HADM_ID).
CALLOUT	Information regarding when a patient was cleared for ICU discharge and when the patient was actually discharged.
CAREGIVERS	Every caregiver who has recorded data in the database (defines CGID).
CHARTEVENTS	All charted observations for patients.
CPTEVENTS	Procedures recorded as Current Procedural Terminology (CPT) codes.
D_CPT	High-level dictionary of Current Procedural Terminology (CPT) codes.
D_ICD_DIAGNOSES	Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD-9) codes relating to diagnoses.
D_ICD_PROCEDURES	Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD-9) codes relating to procedures.
D_ITEMS	Dictionary of local codes ('ITEMIDs') appearing in the MIMIC database, except those that relate to laboratory tests.
D_LABITEMS	Dictionary of local codes ('ITEMIDs') appearing in the MIMIC database that relates to laboratory tests.
DATETIMEEVENTS	All recorded observations are dates, for example, time of dialysis or insertion of lines.
DIAGNOSES_ICD	Hospital assigned diagnoses, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system.
DRGCODES	Diagnosis Related Groups (DRG), which are used by the hospital for billing purposes.
ICUSTAYS	Every unique ICU stays in the database (defines ICUSTAY_ID).
INPUTEVENTS_CV	Intake for patients monitored using the Philips CareVue system while in the ICU, e.g., intravenous medications, enteral feeding, etc.
INPUTEVENTS_MV	Intake for patients monitored using the iMDSoft MetaVision system while in the ICU, e.g., intravenous medications, enteral feeding, etc.
OUTPUTEVENTS	Output information for patients while in the ICU.
LABEVENTS	Laboratory measurements for patients both within the hospital and in outpatient clinics.

MICROBIOLOGYEVENTS	Microbiology culture results and antibiotic sensitivities from the hospital database.
NOTEEVENTS	Deidentified notes, including nursing and physician notes, ECG reports, radiology reports, and discharge summaries.
PATIENTS	Every unique patient in the database (defines SUBJECT_ID).
PRESCRIPTIONS	Medications ordered for a given patient.
PROCEDUREEVENTS_MV	Patient procedures for the subset of patients who were monitored in the ICU using the iMDSoft MetaVision system.
PROCEDURES_ICD	Patient procedures, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system.
SERVICES	The clinical service under which a patient is registered.
TRANSFERS	Patient movement from bed to bed within the hospital, including ICU admission and discharge.

Table 4.1: Tables from MIMIC-III.

Level	**Nodes**	**Total Count**
0	1	891094
1	4	891094
2	73	891094
3	526	891094
4	3585	887782
5	10038	768131
6	7313	249010
7	867	10304

Table 4.2: Codes distribution in MIMIC-III

We hold only the summaries of discharge and their addenda providing the most detailed diagnostic information. The timestamp of the documents gives us the correct reading order for patient admission. Also, some of the hospital stays do not have discharge summaries: following previous studies, we only consider those that do (Perotte et al, 2013; Baumel et al, 2017; Mullenbach et al, 2018). The data set comprises 8,929 unique ICD codes for patients with discharge summaries (6,918 diagnoses & 2,011 procedures). We ignore five of them, however, and use 8,924 codes. Total number of header codes are 1168.

Codes are structurally hierarchal as described in section 2.5, Head code presents the general disease and tail presents specific symptom/disease. Yang et al (2016) worked on the hierarchal level [18]. Our experiment is based on the main codes to find coarse values. It should be remembered that the representation of codes by means of a hierarchy is for convenience purposes only: the only legitimate codes

for diagnosis and billing are the leaf nodes containing the complete codes. No details on the code's hierarchy are included in MIMIC-III, only complete codes are allocated to the nodes. The data set codes are distributed in such a way that most label occurrences are transmitted by a minority of the codes. An unbalanced distribution would significantly decrease the predictive power of the models for rare codes due to a large number of labels. Table 4.2 shows the code hierarchy in detail.

The pre-processed data is split the same way as Mullenbach et al (2018) has done. These splits are patient independent. The summaries of subject discharge can be very long, and this can cause problems as any architecture based on temporal dependencies would fail without long-term memory. The mean number of words per summary is 1505, with a standard deviation of 775. Discharge summaries demonstrate a fixed composition: the note is usually partitioned into segments such as "history of present illness", "social history", "family history", "past medical history", "discharge medications". "hospital course", "discharge diagnosis". We have a fixed maximum size of 5000 tokens in order to avoid very long text.

Code	**Label Description for ICD9 Code**
401.9	unspecified essential hypertension
38.93	venous catheterization not elsewhere classified
428.0	congestive heart failure unspecified
427.31	atrial fibrillation
414.01	coronary atherosclerosis of native coronary artery
96.04	insertion of an endotracheal tube
96.6	enteral infusion of concentrated nutritional substances
584.9	acute kidney failure unspecified
250.00	diabetes mellitus without mention of complication type ii or unspecified type not stated as uncontrolled
96.71	continuous invasive mechanical ventilation for less than 96 consecutive hours
272.4	other and unspecified hyperlipidemia
518.81	acute respiratory failure
99.04	transfusion of packed cells
39.61	extracorporeal circulation auxiliary to open-heart surgery
599.0	urinary tract infection site not specified
530.81	esophageal reflux
96.72	continuous invasive mechanical ventilation for 96 consecutive hours or more
272.0	pure hypercholesterolemia
285.9	anemia unspecified
88.56	coronary arteriography using two catheters

Table 4.3: Top 20 codes in MIMIC-III

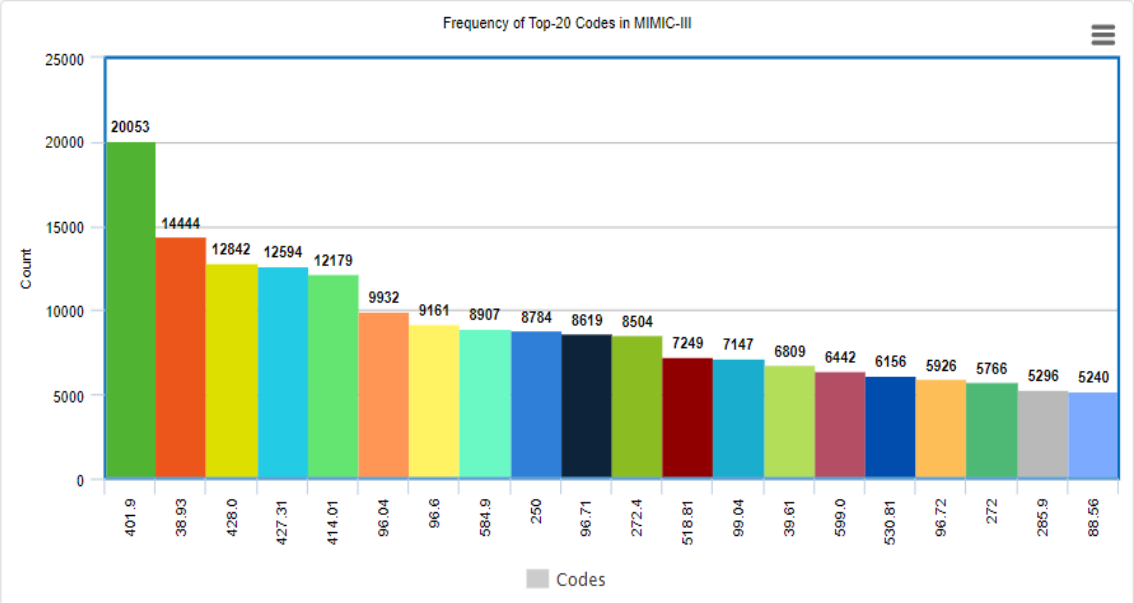

Figure 4.1: Distribution of top-20 codes

Mullenbach et al (2018) [8] performed experiments on top-50 codes that had better distribution. However, each section may contain relevant information: e.g. the patient history section is important because some codes are related to past conditions or procedures (e.g. "personal history of malignant prostate neoplasm" or "personal history of venous thrombosis and embolism"). These codes somehow better present the diagnosis of common illness among patients. table 4.3 shows a list of top-20 ICD-9 labels for the codes displayed in figure 4.1.

4.2 Training & Development details

The model was trained on HPC Polito Server using Pytorch coding structures of neural networks. The training was done on a node of Legion Cluster with one GPU nVidia Tesla V100 SXM2 for dilated convolution neural networks and two CPU Intel Xeon 2.10 GHz for data loading. Gradient optimizer for weight was done with ADAM optimizer (Kingma & Ba, 2014) with initial learning rate 0.0001 $\beta 0 = 0.9$, $\beta 1 = 0.999$, $\epsilon = 1e^{08}$ [50]. The model is designed to save optimum epoch and early-stop training with training criteria. For the reason of the chosen evaluation metric, this criterion is fixed to micro-F1 with the patience of 5.

Neural networks are commonly prone to overfitting, for this reason, various dropouts were tried the best results appeared on 0.15 after many variations. Cross-entropy loss helps to fit the model to targeted labels. After the dropout, each layer performs normalization for smooth convergence. Layer normalization also helps to speed-up training. Data is split into training, validation, and testing using a similar approach by Mullenbech et al (2018) as presented in Table 4.4.

DataSet	Subjects	Total Admissions	Unique Codes
Training	36998	47723	8692
Validation	1374	1631	8929
Testing	2755	3372	3012
Full	41127	52726	4085

Table 4.4: Data Splitting

Numerous training experiments were performed with a batch size of 8 and a maximum token length of 5000. To capture some relevant parameters we performed a trial approach from work done by Mullench et al (2018) and Stefano et al (2019) for Word Embedding dimensions d_w, dropout probability pd, kernel size k, channel size on layers c and dilation length d_L. In table 4.5 below is the selected range of these values for optimal results.

Paramters	Candiate Values	Mellenbach et al,(2018)	Stefano et al (2019)	Our work
d_w	50,100,200,300	200	200	300
p_d	0.1,0.15,0.2	0.2	0.5	0.15
k	3,4,5,6,10	10	3	4
c	100,125,150,200	200	125	125
d_L	3,4,5	-	3	5

Table 4.5: Parametric tuning for the model

CHAPTER 5

Evaluation Metrics

The interpretability of results is of key importance when it comes to the medical domain. Any project requires the evaluation of machine learning models or algorithms. A variety of evaluation approaches for assessing a model are available. One way to give insight into the model's decision-making process is to analyse which sections of the text were considered to be the most important to each document. As current models assign importance to a token in the input document by the label-dependent attention process, analysis of the focus core assigned to various parts of the documents may indicate the linguistic patterns used by the model to create predictions for each document.

For the task of understanding clinical notes, it is important to understand which matrix described the relation between the text and code. Since the distribution of codes across summaries is not balanced. We performed experiments with micro and macro averaged parameters. Micro-averaged values are computed by independently predicting the pairs (texts, codes). The macro-averaged values are determined by combining per-label metrics while less frequently reported in multi-label classification literature.

$$Micro - R = \frac{\sum_{l=1}^{|\zeta|} TP_\zeta}{\sum_{l=1}^{|\zeta|} TP_\zeta + FN_\zeta}$$

$$Macro - R = \frac{1}{|\zeta|} \sum_{l=1}^{|\zeta|} \frac{TP_\zeta}{TP_\zeta + FN_\zeta}$$

Here, TP and FN stand for True positive and False-negative. ICD code classification is usually rare, with most of them labelled false and just a few true, we offer a more detailed quantitative metric of our assessment to the micro-averaged values. The accuracy of a model is how well it captures the portion of correct prediction among all test data points. Precision is a metric that measures the correctness among all positive labels whereas Recall is a metric that measures how many positive labels are successfully predicted amongst all positive labels. Parametric changes make these values fluctuate. A common harmonic mean of these values is known as F-measure. The F1 Score is the simply 2*((precision * recall)/(precision + recall)). the F1 score conveys the balance between the precision and the recall.

The value of the PR-AUC (area under the curve) is determined as the region under the ROC curve for precision, obtained by comparing the true positive rate (TPR) with the false positive rate (FPR) in different thresholds. The ROC AUC value intuitively tests the likelihood that the formula gives a positive instance better than a negative one. The lower bound is 0.5, which is the score obtained by a classifier that classifies the samples as positives with probability 0.5 (a random classifier). ROC AUC tests the actual negative effects which are extremely frequent for this issue, it has the potential to produce very high scores (above 0.9). Next section will compare various scores to the chosen baselines in terms of F1, PR-AUC and P@8.

CHAPTER 6

Results

Our quantitative evaluation is based on finding full ICD-9 codes on MIMIC-III discharge summaries. The results are shown in table 10 with two chosen baseline models Mullenbach et al (2018) [8] and Stefano et al (2019)[43]. The table demonstrates the different results achieved after various experiments. Comparison is being performed with chosen micro evaluation metrics as stated in chapter 5. Our proposed variant of Stefano et al (2019) architecture shows improved results over the state-of-the-art model [43].

We have performed various other kinds of embeddings to observe its effects over coverage and performance. We have made individual coverage tests for Glove, Word2Vec, FastText and stacking them with each other. Mainly, we observed nearly 2 % higher coverage of vocabulary with a different GloVe embedding. This embedding model also expedited learning speed on GPU. Making changes in the embedding layer shows the effectiveness of the model in terms of all evaluation metrics. Out-of-Vocabulary words are initialized in two ways, In the first case for individual pre-trained models, we use random vectors and marked them to become trainable. Additionally, stack embeddings are introduced to cover the gap of OOV words and they perform better in terms of precision but loses the performance in all other metrics. Embeddings with fasttext converge results faster and give full coverage but perform worse than others in terms of evaluation matrices.

Secondly, we use stack embeddings to join results from both embeddings to increase coverage. As compared to Stefano et al (2019) work, there is a noticeable increase in micro-precision, micro-recall, micro-F1 measure on the cost of a decrease in precision area under the curve precision at top 8 codes. These results are computed on coarse evaluation. Model outperformed contrasted to Mullenbach et al (2018) in the PR-AUC score only. Codes for these results can be found at `https://github.com/NeelKanwal/AI-Technologies_For_Clinical_Notes`.

Our model with five dilation layers and GloVe (42B) pretrained embeddings were tested on the various parameters to tune performance. The best results with GloVe embeddings were obtained with kernel size k=3 and dropout probability 0.15, although with variation in these parameters depicted a slight improvement in precision and recall scores. Our proposed model with GloVe delivers (0.772 vs 0.742 vs 0.6322) compared to the work [43] and CAML model [8]. A noticeable rise of almost 3% precision which

shows truly detected codes out of total correct and wrong predicted codes. Additionally, a stacked embedding architecture shows a further hike of 1% in precision relative to other architectures.

Moreover, minor growth can be seen in F-1 and micro-recall (0.661 and 0.595 respectively). These matrices have improved sufficiently at nearly 10% and 6% Mullenbach et al (2018) [8]. We have chosen a potential evaluation metric of precision at the top 8 codes to demonstrate further comparison. The deep dilated model shows a slight decline in this matrix. Table 6.1 outlines all relevant scores and chosen evaluation matrices. PR-AUC is an average of precision score computed for each recall which inherits how better a positive class is chosen. A prime improvement can be seen in this area which is nearly 13%. Our model shows significant development in all areas except precision@8 which is the parameter for the top eight codes only.

Figure 6.1 & 6.2 shows effects on chosen metrics based on filter size and dropout probability respectively. Extending filter size reduces precision on account of longer n-grams that hauls the information from multiple sentences. On the other hand figure, 6.3 & 6.4 visualized the similar outcomes for Word2Vec embeddings. A legend is placed in between to identify the score based on different colors.

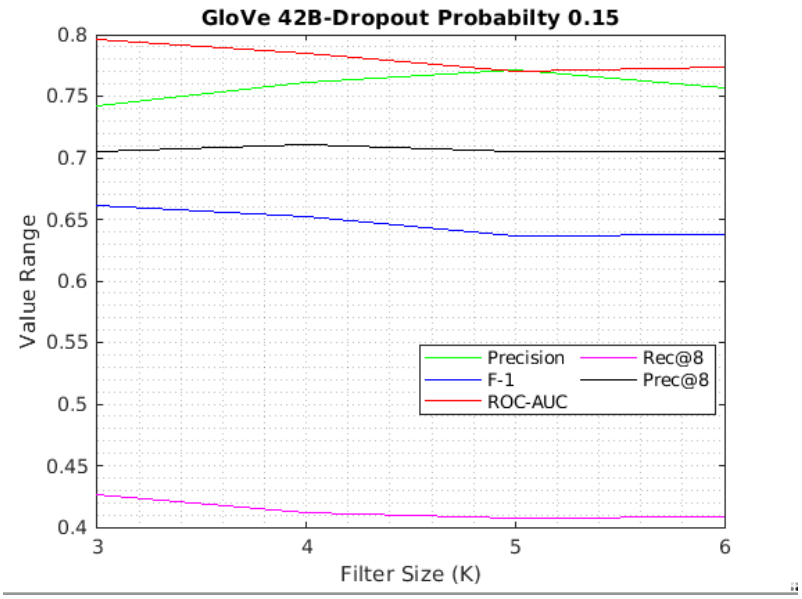

Figure 6.1: GloVe based Embedding with different Kernel Size (dropout=0.15)

Figures 6.5 and 6.6 show the performance of the model with Facebook's fastText embeddings. These embeddings show full coverage based on their mechanism of breaking words but results in an unstable performance in all matrices except precision. It outcomes a slim rise in precision compared to GloVe embeddings after best-chosen parameters. In order to compensate for the performance of other evaluation criteria, we have merged embeddings in a stacked manner with FastText and Word2Vec. Figure 6.7 shows two tables with a change in pre-trained embeddings but performance is lesser to the first model.

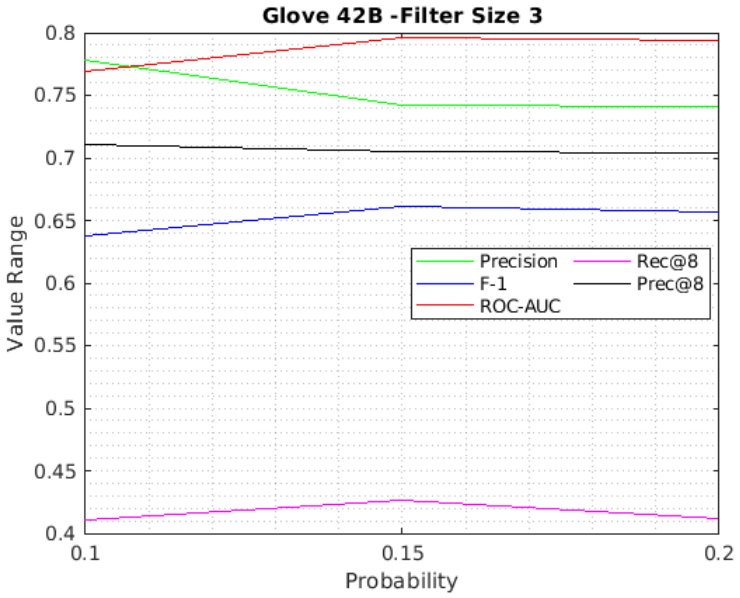

Figure 6.2: GloVe based model with K=3 and various dropout probabilities

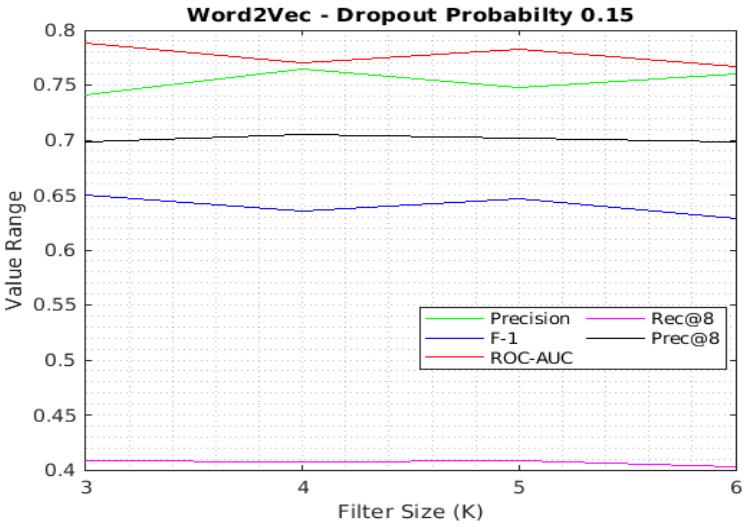

Figure 6.3: Word2Vec Based Model with K=3 and Varying Dropouts

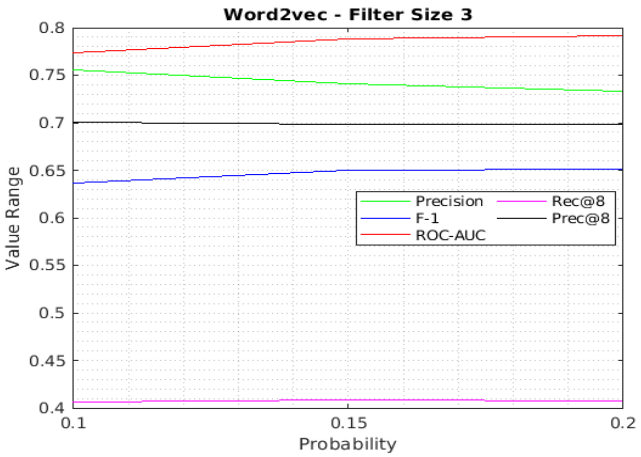

Figure 6.4: Word2Vec based Model with Dropout= 0.1 and varying filter size

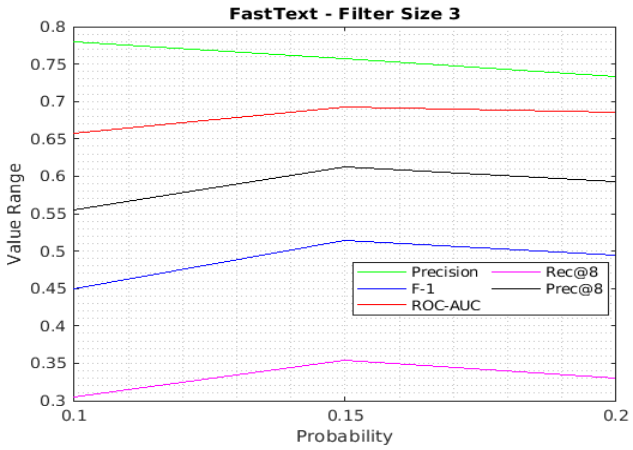

Figure 6.5: FastText Embeddings with Variable Pd

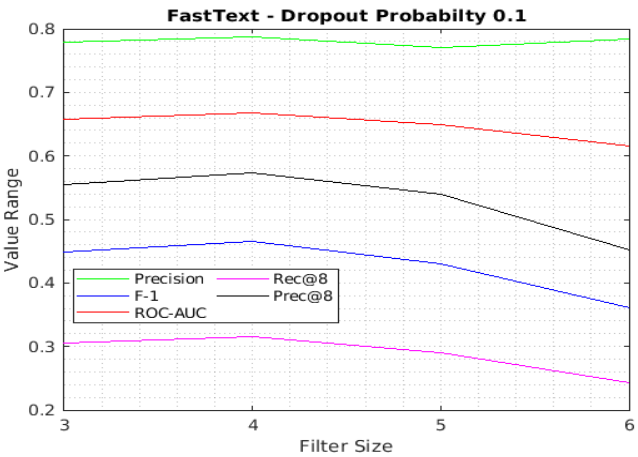

Figure 6.6: FastText Embeddings with variation in filter size

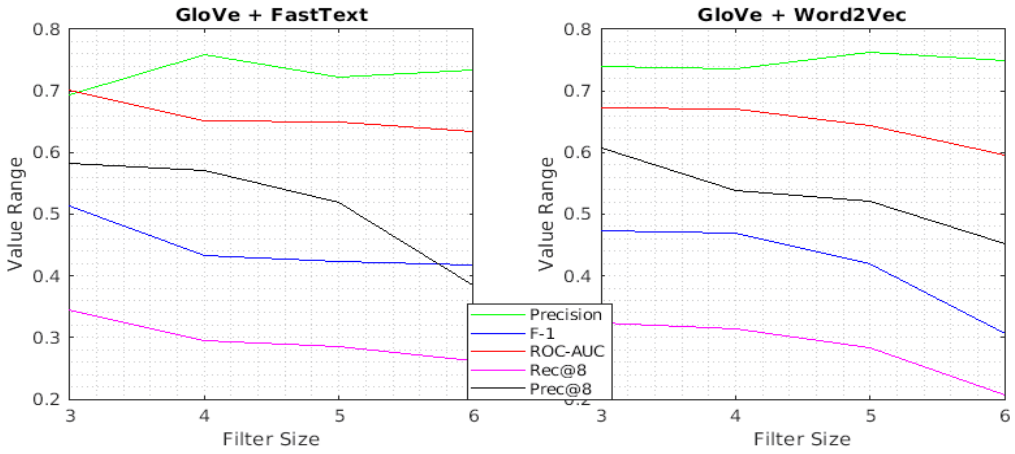

Figure 6.7: Stack Embeddings with different flavours

Architecture	Pre-trained Embedding	Coverage	Micro Precision	Micro Recall	Micro F1	Micro PR-AUC	Prec @ 8
Mullenbach et al (2018)	MIMIC word2vec	1	0.6322	0.442	0.521	N/A	0.69
	Wikipedia Glove	0.5	0.6027	0.491	0.542	N/A	**0.709**
Stefano et al (2019)	Common Crawl Glove(840B)	0.69	0.7427	0.5895	0.6573	0.6689	0.704
Our Model	Word2Vec	0.507	0.7554	0.55	0.637	0.773	0.700
	Common Crawl Glove(42B)	0.715	0.7725	**0.595**	**0.661**	**0.796**	0.704
	Common Crawl Glove(840B)	0.69	0.7427	0.592	0.659	0.794	0.705
	FastText	1	0.757	0.383	0.51	0.693	0.613
	GloVe+Word2Vec	0.74	0.741	0.348	0.47	0.673	0.5829
	GloVe+FastText	1	**0.781**	0.381	0.51	0.689	0.622

Table 6.1: Experimental Results compared to baseline

CHAPTER 7

Conclusion

A large number of categorization and classification schemes are used by the health care system to help with data management for a variety of tasks including patient care, and retrieval, record storage, statistical analysis, insurance, and billing. We have proposed an architecture with deep dilation neural network model for automated clinical notes labeling. The architecture comprises of multiple options of keeping embedding from pre-trained models or keeping a stack from those models jointly. These embeddings are statically formed on word level. The embedding vectors are fed to a five-layer convolution neural network with different spatial filters to understand notes at different levels. The output of this deep dilated convolution is fed further to the dot-based attention layer for label-based attention.

The model is assessed on discharge summaries and procedures of the MIMIC-III v 1.4 dataset. Obtained results here surpasses the existing state of the art model in some evaluation metrics. Preprocessed data is more structures and filter compared to previous studies of shallow CNN and SVM models. We showed how the GloVe 42B model with 300 dimensions gives a better word coverage and improves precision where word2vec based unigram embeddings result in more out-of-vocabulary words.

The model was trained based on the early stopping criteria of the F-1 Score. The training was done a single GPU with 24 gigabytes of RAM. This model has the potentiality of being applied to clinical systems for supplementary assessment. It can deal with a bit of noise data. This model can be pipelined with the data-preprocessing system prior to evaluation.

CHAPTER 8

Future Work

Recent advancement in Natural Language Processing has a huge impact on the deployment of ML algorithms in the healthcare sector. Particularly for medical summaries which can now be better processed than ever before. A significant trust level can be achieved by improving the model to further accurate automated understanding. Distributed models have given a strong foundation for next-generation Biomedical Natural Language Processing (Bio-NLP). Several propositions can be presented for this work.

First development can be improving word embeddings from statical embeddings to dynamic (Context Level Embeddings). These dynamic embedding can create a strong semantic understanding of words that have a different meaning in different sentences. Moreover, embeddings can be created with newer models like BioWord2Vec or Biological Encoder Representation from Transformers (Bio-Bert) which are medical terminology oriented. These models can also reproduce lower-dimensional embeddings which may result in faster processing and better accuracy.

Secondly, Transformer based models can be implemented for similar tasks on account of its diversity in NLP tasks. It has encoder-decoder multi-head attention layer [2]. These attention scores can be used to extract most important sentences in summaries. Seq2seq models have shown great potential in machine translation and can deal with longer dependencies. In clinical notes, prescriptions are sometimes longer where some sentences deal with disease identification. This matter can be better understood with a sentence level self-attention mechanism. Domain specialized Transformer models like Clinical Bidirectional Encoder Representation from Transformers (Clinical Bert or Medical Bert) can produce performance enhancements on common clinical NLP tasks respect to nonspecific embeddings [4].

Most of the research in clinical NLP so far has been carried out on the semantic parsing of patient notes. Forming a state-of-Art Model for a long text-document with the characteristic of the healthcare domain is pipe-lining work. Especially, dealing with many technical words and typos/misspellings is tricky at first hand. Upcoming fine-tuned transformer models for the medical domain may totally automate a clinician job.

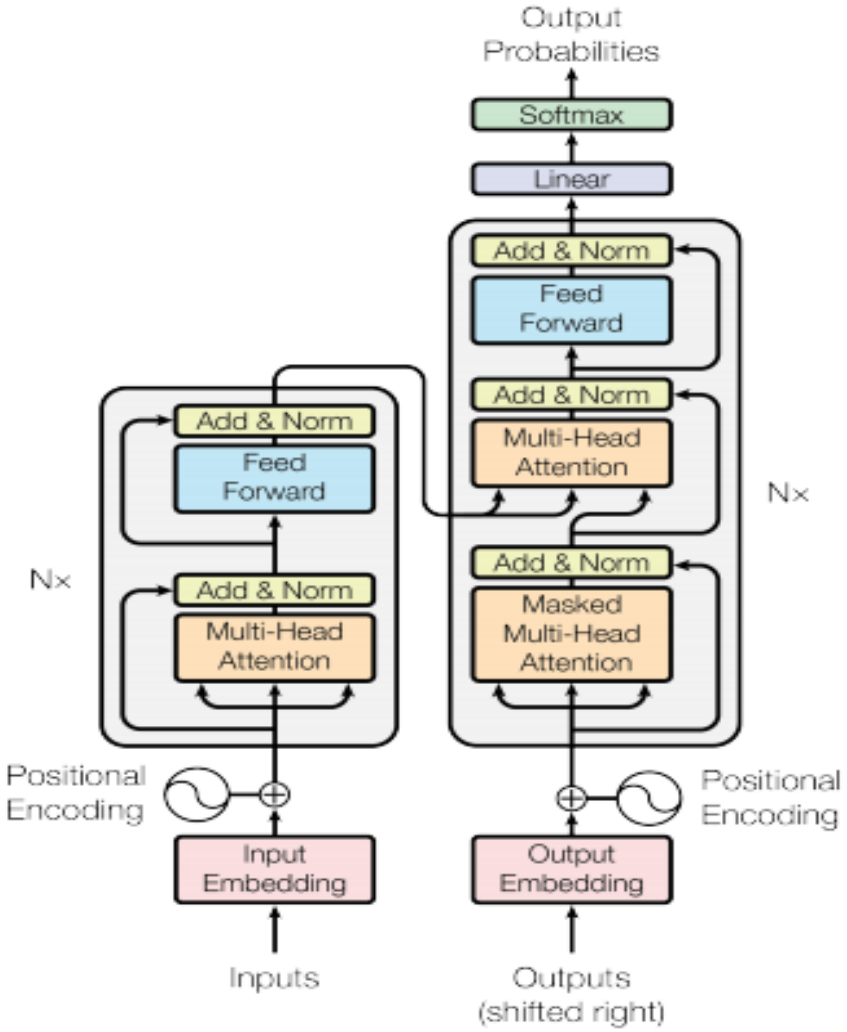

Figure 8.1: Transformer Model (Ashish et al 2017[2])

Bibliography

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, 2019, Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar et al, 2017, Attention is all you need, Advances in Neural Information Processing Systems 30 (NIPS 2017)

[3] Matthew E. Peters, Mark Neumann, Mohit Iyyer, 2018 Deep contextualized word representations, arXiv:1802.05365

[4] Emily Alsentzer, John R. Murphy, Willie Boag, 2019, Publicly Available Clinical BERT Embeddings, Clinical Natural Language Processing (ClinicalNLP) Workshop at NAACL 2019, arXiv:1904.03323

[5] Jeffrey Pennington, Richard Socher, Christopher D. Manning, 2014, GloVe: Global Vectors for Word Representation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Pages: 1532–1543

[6] Armand Joulin et al, 2016, Bag of Tricks for Efficient Text Classification, arXiv Preprint, arXiv:1607.01759

[7] Piotr Bojanowski et al, 2016, Enriching Word Vectors with Subword Information, Transactions of the Association for Computational Linguistics, Volume 5 pages: 135-146

[8] James Mullenbach et al, 2018, Explainable Prediction of Medical Codes from Clinical Text, Computational Linguistics: Human Language Technologies, Volume 1, Pages: 1101–1111

[9] Edward Choi et al, 2016, Doctor AI: Predicting Clinical Events via Recurrent Neural Networks, arXiv Preprint,arXiv:1511.05942

[10] Birman-Deych E et al, 2005, Accuracy of ICD-9-CM codes for identifying cardiovascular and stroke risk factors, PubMed, DOI: 10.1097/01.mlr.0000160417.39497.a9

[11] Tal Baumel et al, 2017, Multi-Label Classification of Patient Notes a Case Study on ICD Code Assignment, In AAAI Workshop on Health Intelligence, arXiv Preprint, arXiv:1709.09587

[12] Richárd Farkas, György Szarvas, 2008, Automatic construction of rule-based ICD-9-CM coding systems, BMC bioinformatics 9(3): S10

[13] Crammer et al, 2017, Automatic code assignment to medical text, In Proceedings of the ACL Workshop on BioNLP 2007

[14] Lipsky Gorman et al, 2010, Section classification in clinical notes using supervised hidden Markov model, In Proceedings of the 1st ACM International Health Informatics Symposium, 744–750. ACM

[15] Tomas Mikolov et al, 2013, Efficient Estimation of Word Representations in Vector Space, In Proceedings of the International Conference on Learning Representations (ICLR 2013)

[16] Perotte et al,2014, Diagnosis code assignment: models and evaluation metrics, in JAMIA 2014, DOI:10.1136

[17] Pestian et al, 2007, “A shared task involving multi-label classification of clinical free text”. In Proceedings of the ACL Workshop on BioNLP: Biological, Translational, and Clinical Language Processing, 97–104.

[18] Yang et al, 2016, Hierarchical attention networks for document classification. In Proceedings of NAACL-HLT, 1480–1489

[19] Shi et al, 2017, Towards automated ICD coding using deep learning. arXiv preprint arXiv:1711.04075.

[20] Ramakanth Kavuluru et al, 2017, An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. Artificial intelligence in medicine 65(2):155–166.

[21] Volodymyr Mnih, Nicolas Heess et al, 2014, Recurrent Models of Visual Attention, Recurrent Models of Visual Attention, Pages 2204-2212

[22] Kim Yoon, 2014, Convolutional neural networks for sentence classification, In Conference on Empirical Methods in Natural Language Processing

[23] Johnson, Alistair E. W., Pollard, Tom J., Shen, Lu, Lehman, Li-wei H., Feng, Mengling, Ghassemi, Mohammad, Moody, Benjamin, Szolovits, Peter, Celi, Leo A., & Mark, Roger G. 2016. MIMIC-III, a freely accessible critical care database. Scientific Data, 3(May), 160035+

[24] Jean-Baptiste et al, 2018, Deep Representation for Patient Visits from Electronic Health Records, arXiv Preprint, arXiv:1803.09533

[25] Minh-Thang Luong, Hieu Pham, Christopher D. Manning, Effective Approaches to Attention-based Neural Machine Translation, 2015, arXiv preprint, arXiv:1508.04025

[26] Miotto R, Li L, Kidd BA, Dudley JT. Deep patient: an unsupervised representation to predict the future of patients from electronic health records. Scientific reports. 2016 May 17;6:26094.

[27] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio,Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, 2015, arXiv preprint, arXiv:1502.03044

[28] Henry, J., Pylypchuk, Y., Searcy T. & Patel V. (May 2016). Adoption of Electronic Health Record Systems among U.S. Non-Federal Acute Care Hospitals: 2008-2015. ONC Data Brief, no.35. Office of the National Coordinator for Health Information Technology: Washington DC.

[29] Sanjay Purushotham et al, 2017, Benchmark of Deep Learning Models on Large Healthcare MIMIC Datasets, arXiv Preprint, arXiv:1710.08531

[30] Lita, L. V.; Yu, S.; Niculescu, R. S.; and Bi, J. 2008. Large scale diagnostic code classification for medical patient records. In Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), 877–882

[31] Cho, Kyunghyun, Aaron Courville, and Yoshua Bengio. “Describing Multimedia Content using Attention-based Encoder-Decoder Networks. “ arXiv preprint arXiv:1507.01053 (2015).

[32] Itti, Laurent, Christof Koch, and Ernst Niebur. “A model of saliency-based visual attention for rapid scene analysis. “IEEE Transactions on Pattern Analysis & Machine Intelligence 11 (1998): 1254–1259.

[33] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate. “ arXiv preprint arXiv:1409.0473(2014).

[34] Wenpeng Yin et al, 2017, Comparative Study of CNN and RNN for Natural Language Processing, arXiv preprint, arXiv:1702.01923

[35] Marc Moreno Lopez and Jugal Kalita, 2017, Deep Learning applied to NLP, arXiv preprint, arXiv:1703.03091

[36] Mark HUGHES et al, 2017, Medical Text Classification using Convolutional Neural Networks, Studies in health technology and informatics, DOI: 10.3233/978-1-61499-753-5-246

[37] Fisher Yu, 2016, MULTI-SCALE CONTEXT AGGREGATION BY DILATED CONVOLUTIONS, arXiv preprint, arXiv:1511.07122

[38] Gavneet Singh Chadha, Jan Niclas Reimann, Andreas Schwung, 2019, Generalized Dilation Neural Networks, arXiv preprint, arXiv:1905.02961v1

[39] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals (2003). Circulation. 101(23):e215-e220

[40] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov, Enriching Word Vectors with Subword Information, 2016,arXiv,arXiv Preprint: 1607.04606

[41] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998. doi: 10.1109/5.726791

[42] A. Waibel, "Prosodic knowledge sources for word hypothesization in a continuous speech recognition system," ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing, Dallas, TX, USA, 1987, pp. 856-859. doi: 10.1109/ICASSP.1987.1169848

[43] Stefano Macrino , Carmelo Velardo , Giuseppe Rizzo , "A Dilated CNN Approach for Coding of Clinical Notes", Submitted to Transactions on Computing for Healthcare, 2020,

[44] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan, Show and Tell: A Neural Image Caption Generator, 2014, arXiv preprint, arXiv:1411.4555

[45] Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG, MIMIC-III, a freely accessible critical care database, 2016,DOI: 10.1038/sdata.2016.35.

[46] Rajaraman, A.; Ullman, J.D. (2011). "Data Mining" (PDF). Mining of Massive Datasets. pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 978-1-139-05845-2

[47] Nils J. Nilsson, INTRODUCTION TO MACHINE LEARNING,1998,Robotics Laboratory Stanford University

[48] L. P. Kaelbling, M. L. Littman, A. W. Moore, Reinforcement Learning: A Survey, 1996,Journal of Artificial Intelligence Research, Vol 4, (1996), 237-285, arXiv:cs/9605103

[49] World Health Organization (WHO), https://www.who.int/classifications/icd/en/

[50] Computational resources were provided by HPC @ POLITO (http://www.hpc.polito.it)"