Bauhaus-Universität Weimar
Faculty of Media
Degree Programme Computer Science for Digital Media

Collaborative Semi-supervised learning Framework for Clinical NLP

Master's Thesis

Sebastian Laverde Alfonso Matriculation Number 119414

1. Referee: Prof. Dr. Benno Stein

Submission date: March 3, 2022

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Weimar, March 3, 2022

.....
Sebastian Laverde Alfonso

Abstract

Modern natural language processing (NLP) provides promising methods for transforming healthcare in areas like drug discovery, patient care, consultation and diagnosis. However, experimental data is scarce and remains costly, primarily due to its sensitivity, foreclosing computational models from being useful in practice. Here, the author proposes a semi-supervised collaborative framework capable of solving several clinical NLP tasks such as automatic de-identification and classification of electronic medical records, by performing controlled text generation with little supervision, where a set of tunable and disentangled aspects, condition the featural profile of generated fake records, while maintaining the integrity of the data. The data used is from i2b2 NLP challenges for de-identification and smoking status classification, and consist of 889 and 502 medical discharge summaries respectively. Previous solutions, overfit on syntactic particularities of the training data and resembles named-entity recognition, which make them good inference tools only for in-domain data but have difficulties extrapolating and are far from being scalable general solutions for these or more NLP tasks in Healthcare. The model used is a variational auto-encoder (VAE) with holistic attribute discriminators as proposed by Hu et al., 2018 [27], which through collaborative learning, learns interpretable representations and produces surrogate records with the desired presence-level of a set of attributes like personal healthcare information and smoking status information. The VAE is based on gated recurrent units and the discriminators are text convolutional neural networks. A model like this, is capable of doing data augmentation, controlled text generation, and carrying out automatic classification and interpolation of data points. These results demonstrate that a framework based on VAEs as flexible and scalable like this, boosted with extra regularization and latent space enrichment techniques, is highly advantageously with potential general application to data scarce NLP learning tasks on healthcare.

Contents

List of Figures

Acknowledgements

To my parents for laying the tracks, for their unconditional love and support.
To the Webis Group and the DLR for the opportunity and all the learning.
To the data annotators, data providers and the Internet for making it all possible.
Special thanks to Michael Voelske and Tim Gollub for all their help and great attitude.

Chapter 1

Introduction

1.1 Motivation

The first thoughts about *Artificial Intelligence* (AI) can be traced back to the first attempts to describe human reasoning as a symbolic procedure. With latest advancements in computer science and our understanding of biological thinking processes, we can now develop computer systems able to mimic human behaviour and perform tasks that would usually require human intelligence, such as understanding text. Particularly, state-of-the-art *Deep Learning* (DL) models, where the algorithms[1] make use of brain-like logical structures called *Artificial Neural Networks* (ANN), present remarkable applications in language understanding.

Natural Language Processing (NLP) refers to the branch of computer science, and more specifically, the branch of AI concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.[2] Figure 1.1[3] shows a Venn's diagram of these related fields. Now, *Clinical NLP* refers to these computational techniques applied to textual sources—such as medical records—containing health information about a patient.[4]

Health information includes past, present, and future information about mental and physical health and the condition of an individual, the provision of healthcare to an individual, and information related to payment for healthcare,

[1] series of instructions telling a computer how to solve a problem or task. This usually means transforming a set of facts about the world (data) into useful information (concepts).

[2] taken from *IBM Cloud Education 2020*.
`https://ibm.com/cloud/learn/natural-language-processing`

[3] inspired by Skynet [58]

[4] common clinical textual sources: surgical and pathology reports, clinical discharge summaries and progress notes

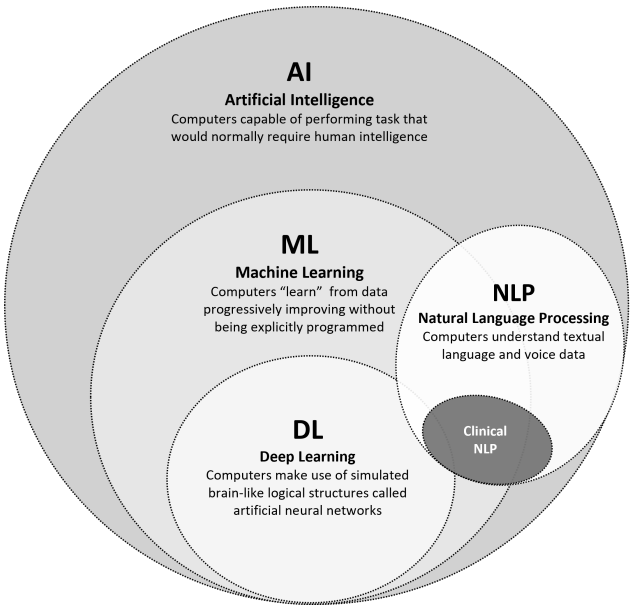

Figure 1.1: Venn Diagram portraying how AI, ML, DL, NLP are related.

again in the past, present, or future. Health information also includes demographic information about an individual and information or opinion about an individual's illness, injury or disability [2].

The *Health Insurance Portability and Accountability Act* or HIPAA is a federal law imposed on 1996 that required the creation of national standards to protect sensitive information from being disclosed without the patient's consent or knowledge [1]. This information is known as *Individually Identifiable Health Information* (IIHI) and constitutes the health information created or received by a healthcare provider, health plan, employer, or healthcare clearinghouse, that identifies the individual, or with respect to which there is a reasonable basis to believe the information can be used to identify the individual [49].

Now, the *HIPAA Privacy Rule* is a set of standards that protects most IIHI held or transmitted by a covered entity[5] or its business associates[6], in any form or medium, whether electronic, on paper, or oral [50]. This is know as *Protected Health Information* or PHI. If a HIPAA-covered entity has a data set containing IIHI, the data must first be de-identified previous to sharing the information with an organization or individual. The Privacy Rule provides

[5]a health care provider that conducts certain transactions in electronic form. For instance, a health care clearinghouse, which a third-party system that interprets claim data between provider systems and insurance payers, or a health plan [1].

[6]a person or entity that performs certain functions or activities that involve the use or disclosure of protected health information on behalf of, or provides services to, a covered entity.

two de-identification methods: a formal determination by a qualified expert in statistics or scientific principles; or the removal of specified individual identifiers as well as absence of actual knowledge by the covered entity that the remaining information could be used alone or in combination with other information to identify the individual. The latest is known as the *Safe Harbor* method and is the focus of automatic de-identification methods.

For clinical data to be considered de-identified, the *Safe Harbor* method requires 18 PHI identifiers to be removed. Figure 1.2[7] shows the identifiers of the individual or of relatives, employers, or household members of the patient, that need to be removed for a medical record to be considered as de-identified.

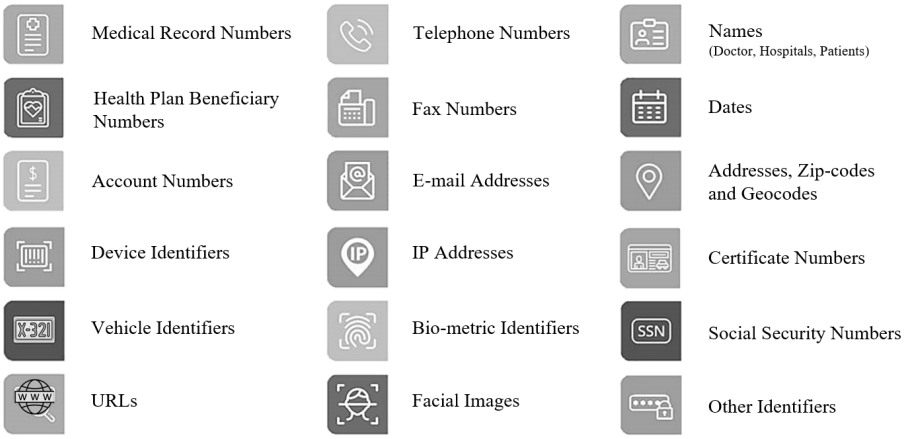

Figure 1.2: HIPAA PHI categories

De-identification ensures the removal of identifiers (PHI) that directly or indirectly point to a person (or entity). In practice, this process has to be done while maintaining the integrity of the data[8], enabling agencies to collaborate in research efforts and produce better services, while preserving the patient's or entity's privacy. There is a strong trade-off between the privacy of the data defined by these identifiers, and utility defined by the value of the transformed data. This is especially important in healthcare, where a very good model could significantly help in the diagnosis of a mortal disease, but where the data is extremely sensitive, ergo scarce.

Medical records can be an important source of information for clinical and laboratory researchers alike, and the potential for clinical natural language processing research is vast. Deep learning is already being used for early-detection diagnosis, normal and end-of-life treatment, drugs discovery, and also to keep

[7]inspired by https://www.bridgepatientportal.com/blog/Protecting-Telehealth-Patient-Data-With-HIPAA-Compliant-Video-Conferencing/

[8]with a similar distribution to the source; retaining key medical concepts

people healthy via tech-apps, just to name some real-life uses. These applications leverage patterns found in data, and are usually *data-hungry*, meaning that the more data, the better. Nevertheless, hospitals and patients are not eager to share their data because of privacy concerns: someone's personal identity (or relatives) from being revealed. So, most of the textual data in this domain has to be carefully de-identified before making used of it as input for training NLP models.

Now, most of the information in medical records comes in the form of free text. Identifying and removing certain arbitrary information is easy to do manually, but at a very high cost: you need medical experts and statisticians to check each individual record. That's the reason why public data in this domain is very scarce and depending on the de-identification method, sharing it can still present a security risk for the patients and data owners. This means that in practice, creating big data sets of de-identified medical records to extract useful information from them, requires automatic processing. The construction of automated systems capable of de-identifying clinical documents is nowadays an important challenge for the research community in clinical NLP; nevertheless, current state-of-the-art in automatic de-identification shows that these systems could and should be better. This is mainly because they strongly depend on the availability of the data and its local distribution, impeding them to generalize well to *out-of-domain* data.

Optimal strategies to automatically de-identify clinical documents would also facilitate the availability of clinical narratives for *Information Retrieval* (IR) applications, which are of prime importance for clinical research [45]. Although the potential uses of IR from clinical text are numerous and far-reaching, most applications rarely occur outside of the local scope, mostly because of scalability issues, which could be overpassed if large sets of de-identified data were made available. This represents a strong motivation to develop data augmentation and automated de-identification systems [19].

A number of investigators have developed methods for automatically de-identifying medical records by removing PHI, as specified in the HIPAA *Safe Harbor* method but they are far from being general solutions, due to their high dependency on the training data and because they are too task-specific, hence not useful for other NLP tasks like classification, such as Uzuner et al. [60] and FerrÃ¡ndez et al. [19]. Consequently, continuous efforts on unsupervised ML methods with state of the art language models that can surpass these problems, is currently an important line of research in clinical NLP. More information about the related work can be found on chapter 2.

Considering that the creation of *gold-standard* data sets is very time consuming and requires access to sensitive data, it is difficult to increase the size a considerable amount (> 1.000). Uzuner et al. [60], anticipated that unsuper-

vised techniques, fed from de-identified data created by models such as theirs, will need to be developed. It is clear then that in order to deal with data scarcity and to create highly scalable general models for NLP in healthcare, the problem has to be formulated and tackled differently.

In this work, instead of thinking the problem of de-identification solely as a discriminative task, where the system identifies PHI in the token level[9] and instance level[10] [60], the author's idea is to generate a novel or an surrogate version of a medical record, controlling the generative process by injecting codes with featural profiles to the *latent representations*[11], so the output is a surrogate record with a minimum content of PHI and maximum content of smoking status information about the patient. This is known as *Controlled Text Generation* (CTG) and it is an application of *Conditional Language Modelling*[12]. Current advances in language modelling and CTG show that is possible to insert desired semantic and syntactic attributes to the generated text, via conditioning while the model is learning. The control mechanism is usually applied to featural information like sentiment and tense of to generate short sentences, but it can also be applied to generate longer pieces of text, such as news articles, movie scripts and medical records, and to control arbitrary aspects of the text. Additionally a system with generative a discriminative capabilities like this, could be use to solve both de-identification and classification tasks with a single pipeline, leveraging patters found in the original data that and the fake generated records in a semi-supervised manner.

Under this thought, the author solves automatic de-identification of medical records and automatic classification of patient's smoking status, with a multi-task model capable of doing controlled text generation with scarce training data, where the controlled aspects are initially the presence or not of personal healthcare information (PHI) or smoking status information in the medical records.

The data sets used in this work are from i2b2 2006 NLP challenges for automatic de-identification and smoking status classification, and consist of 889 and 502 medical discharge summaries respectively [60][61]. Each of these previously de-identified records is labelled whether with one of 4 smoking status categories or contains PHI tags from 8 categories as stated on the challenge.

[9] correcting classification of words in a PHI category. A token is a word or string of contiguous characters between two spaces, or between a space and punctuation marks

[10] instance-level evaluation checks individual PHI instances and marks the presence of a correct instance or one of three types of errors: substitution, insertion, or deletion [65]

[11] latent variables are variables that cannot be measured directly and therefore have to be inferred from the empirical measurements. examples include variables like pain, satisfaction, abilities to perform activities of daily living, stress, burnout or well-being, and health [33]

[12] assigns probabilities to a set of words, given some code or context c. The code c provides a point of control over the generation process.

Here the author focused his work on 3 of the most sensitive PHI: the names of **patients**, **doctors** and **hospitals**, and the smoking status label for each record. The data is transformed to produce 3 experimental data sets, 2 from the smoking status classification challenge (with and without headers[13]) and 1 from the de-identification challenge (without headers), with binary labels as follows:

- **Present:** Attribute i is present in text x.
- **Not present:** No sign of attribute i in text x.

Where the text x is a single medical record (or part of it) and the attributes are personal healthcare information and smoking status of the patient. Further description of the data can be found on chapter 3.1.

This present a fundamental difference with the challenges submissions, where the goal of one is to identify the tokens or entities in text that correspond to PHI, and the goal of the other is simply text classification, because here the author identifies globally if the text contains PHI or smoking status information and then generates a surrogate where the featural profile is controlled, for example, to exclude PHI or smoking status content from it. A more detailed description of the challenges, the distribution of the data, pre-preprocessing and the can be found on chapter 3.

The model's pipeline is denoted here as CVAE-TextCNN to process medical records in XML format, and train a conditional variational auto-encoder (CVAE) combined with attribute discriminators (TextCNNs), based on Hu et al. [27], that imposes desired semantic structures on novel generated medical records. This model, together with regularization techniques and fine-tuning is capable of performing both generative and discriminative task solving de-identification and classification with little supervision. Figure 1.3[14] illustrates the processes where we can distinguish generative and discriminative roles.

Additionally to the pipeline, the author of this work lays the tracks to further improvement of CVAE-TextCNN, refine coherence of the generate text, and to regularize the latent space representation of the medical records, as further steps to create a more robust framework.

1.2 Structure of the Thesis

Chapter 2 present the bibliographical background and state-of-the-art implementations for handling free text on medical records for generative and dis-

[13] a header is a title or section inside a patient's medical record text, that refers to a 'field' or to the 'topic' that the medical information refer to.

[14] original pipeline proposed by Hu et al. [27]

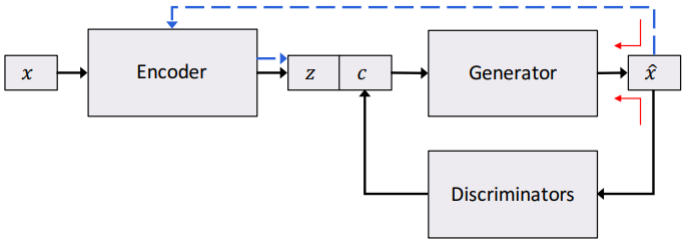

Figure 1.3: Controlled text generation VAE with feature discriminators.

criminative task, it describes the current approaches paradigms and their difficulties, and lays the base for regulated latent representation learning (LRL) systems and the methods for performing controlled text generation.

The data used for this work is described in Chapter 3, including its posterior transformation for creating 3 data sets It also compiles a detailed description about the design decisions taken for this work. Explains how the system works, how it is trained and evaluated. At the end of this chapter, the author present the output files and licensing. The requirements for reproducing the results and using the model and repository are also described there.

The experiments descriptions and results are presented in Chapter 4, together with several figures and tables with common metrics that guided the analysis of performance of the model.

Some considerations about the objective and results of this work, scalability of the presented system and practical implications can be found on Chapter 5, together with author's last thoughts about an accurate general model for controlled text generation being used as a block in a bigger network for healthcare, and the conclusions for this work.

Chapter 2

Related Work

Typically in clinical NLP, the de-identification tools are classified as ruled-based systems or machine learning systems. Rule-based systems usually tackle the de-identification task leveraging syntactic information using pattern matching, regular expressions and dictionary lookups. Although these systems are easy to develop, they very fast and do not require any labelled data, the need for experienced domain experts that can occur and to manually write dictionaries and patterns—catching all possibilities—for each dataset, make them hard to generalize [19].

Conversely, ML systems are classified as supervised or unsupervised depending on the use or not of labelled data. Supervised ML systems are inference models to train classifiers, where in the case of de-identification, each word or instance is labelled as PHI or not. These systems could end up being really efficient and more general than ruled-based ones at the cost of transparency; nonetheless, they depend on features handcrafted by someone with medical and statistical knowledge the same way rule-based system depend on the expert, making the task similarly challenging and time-consuming [4].

Recent approaches in NLP tasks such as *Named-entity Recognition* (NER)[1] and *Parts of Speech* (POS) tagging [2] using non-linear neural networks have shown promising results without any handcrafted features or rules. The features in these systems are learned automatically with other parameters of the network during training on a labelled dataset. Yet, de-identification differs from NER in its focus on clinical records in one important way: the goal in the challenge as reported by Uzuner et al. [60] is to find and remove PHI from

[1] subtask of Information Retrieval that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities.
`https://en.wikipedia.org/wiki/Named-entity_recognition`

[2] categorizing words in a text corpus in correspondence with a particular part of speech (noun, verb, adjective,..)

medical records while protecting the integrity of the data as much as possible. The systems need to achieve this goal in the presence of ambiguities and out-of-vocabulary words.

Between these two approaches there also have hybrid systems and ML methods based on regular expressions templates. Particularly of 2006's NLP challenge for de-identification, statistical ML methods based on regular expressions perform the best between these approaches, followed by hybrid systems combining rules for some PHI categories with learning for others, pure learning systems without regular expression template features or supplementary rules, and pure rule-based systems (Uzuner et al. [60]) almost all used patterns, e.g., regular expressions or rules, for recognizing the format of the PHI. In general the de-identification approaches in 2007 viewed the task as a problem of classification of tokens. Others viewed it as a sequence tracking problem using *Hidden Markov Models* (HMM) (Manning and SchÃ¼tze [42]and i corporation [28]) or *Conditional Random Fields* (CRF) (Lafferty et al. [35], Aramaki et al. [5], Wellner [63] and Wellner et al. [64]). For example Aramaki et al. [5] uses CRFs to learn the features that are important to identify PHI. They take a *text-chunking* and *sequence-tracking* approach to de-identification and mark all tokens as either beginning a chunk or as being inside or outside of a chunk using *IOB tagging* (Sang and Veenstra [55]).[3] These implementations leverage local properties of the data—such as positions of PHI tags in the text—making them too specific. Additionally, some ambiguous and out-of-vocabulary PHI cause missed or partially recognized PHI. In general these approaches are cheap and easy to implement, do not not scale well with other data and depend on the quality of the data (annotation procedures).[4] Extrapolation is difficult when the systems take advantage of the specific structure of discharge summaries, which is characteristic of the entity from where these were collected.

Latent Representation Learning, GANs and VAEs

Other approaches make use of unsupervised learning algorithms, to identify hidden patterns in unlabelled input data. *Unsupervised* refers to the ability to learn and organize information without providing an error signal to evaluate the potential solution. The lack of direction for the learning algorithm in unsupervised learning can sometime be advantageous, since it lets the algorithm to look and find hidden patterns that have not been previously considered. When these hidden patterns or variables cannot be measured directly because

[3]IOB (inside, outside, beginning) is a common tagging format for tagging tokens in a chunking task in computational linguistics.
`https://en.wikipedia.org/wiki/Inside--outside--beginning_(tagging)`

[4]as reported by Uzuner et al. [60]

they don't correspond to any explicit featural information of the input, they are called latent and they have to be inferred [53]. *Latent Representation Learning* (LRL), or *Latent Variable Modeling* (LVM), is a machine learning technique that attempts to infer latent variables from empirical measurements (Kopf and Claassen [33]). And creating latent space representations is something deep neural networks are very good at, solving tasks inside generative modelling that are useful for this work.

The essential idea of generative models is to approximate the underlying data distribution by training a model to fit the training data, finding and leveraging the dependencies and correlations between units (words, sentences, etc), to then build models capable of generating observable data values. The way the model finds this information depends on what it is intended to do with the model and how we define the loss function. Several models have been proposed, such as *Latent Dirichlet Distribution*, *Restricted Boltzmann Machines*, *Generative Adversarial Networks* (GANs) and *Variational Auto-Encoders* (VAEs) [33]. The latest two are the baseline networks for deep learning generative state-of-the-art applications. These models make use of a discriminator and a generator but differ particularly in the way they compute the loss function.

The core idea of GAN is to play a *min-max game* between a discriminator and a generator, known as *adversarial training*. The discriminator tries to differentiate between real samples and surrogates, while the generator tries to create realistic samples that can trick the discriminator into classifying them as real (Li et al. [40]). GANs replace the *maximum likelihood*[5] in training to simulate the real data distribution and generate *high-quality* text.[6] However, the original GAN is only suitable for processing continuous data such as images, while text is discrete data; hence, it cannot be applied directly to text generation (Guo et al. [24]). The details of how GANs work is not the scope of this work; VAEs are generally more suitable for text and more controllable than GANs when including conditional codes so it's the centre of attention of the research of this work.

VAEs are called *autoencoders* (AE) only because they resemble a traditional autoencoder, but an AE is solely trained to encode and decode with as few loss as possible, no matter how the latent space is organised. The VAE on the other hand, instead of encoding an input as a single point, encodes it as a distribution over the latent space, meaning unlike AE, its training is regularised to avoid overfitting and ensure that the latent space has good

[5]method of estimating the parameters of an assumed probability distribution, given some observed data.
`https://en.wikipedia.org/wiki/Maximum_likelihood_estimation`

[6]coherent and similar to the input; looking like real data.

properties. Moreover, the term *variational* comes from the close relation there is between the regularisation and the variational inference method in statistics [54]. in Chapter 3.

The work of Bowman et al. [10] for instance, is an RNN-based VAE text generation model which assigns distributed latent vectors to whole sentences. *Recurrent Neural Networks* (RNNs) are networks design to deal with sequence modeling. A sequence autoencoder model constructs the output sentence word by word conditioned on the hidden vector to obtain consistency and diversity. Notwithstanding RNNs face gradient propagation issues that prevent the model from effectively capture long-term dependencies.

Attention mechanism networks like the *transformer* model are proposed to replace the sequential structure RNN-based models. The self-attention mechanism can capture the context dependencies among all sequences and words, to achieve more efficient sequence modeling without distance restrictions and to obtain more semantically-rich text representations. Transformers have shown excellent performance in various NLP tasks and are currently used in some of the best language models so far like GPT3 (OpenAI, 2020), the Switch Transformer (Google 2021), Megatron (Microsoft and Nvidia, 2021) and Gopher (DeepMind, 2021). It has great development and potential too for the de-identification task [4]. For the purpose of this work, the inclusion of transformer models plays as an extension to the encoder/decoder architectures; the focus is on a cheap yet functional framework capable for solving clinical NLP tasks including automatic de-identification and classification.

Attention Mechanism

Attention is a technique in neural networks that enhances some parts of the input data while diminishing others, meaning that the network devotes more focus—ergo attention—to that small but important part of the data.[7] Ahmed et al. [4] de-identify textual data based on the self-attention mechanism and a stacked RNN. They compare experimental results of architectures based on RNNs with others using *Gated Recurrent Units* (GRU) and analyse the effects of conditional random fields and applying attention mechanism. Their results prove that attention-based models perform better and with a faster executing time for de-identifying MIMIC-III [8] and i2b2 datasets. Yet, an RNN-based model looks at the tokens sequentially, whereas the attention-based model looks at the whole sentence at the same time and process it. This means

[7] based on: `https://en.wikipedia.org/wiki/Attention_(machine_learning)`

[8] freely-available database comprising de-identified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012.

that a GRU architecture could deal better with back-propagation [9] issues like vanishing gradients with less complexity. On the side, they evaluate the quality of the de-identified surrogate records, by introducing utility metrics based on BLEU scores,[10], which reveals how similar two sentences are in a document, and topic modelling methods using *Latent Dirichlet Allocation* (LDA), which is an unsupervised machine learning algorithm that clusters relevant words to a topic (Ahmed et al. [4] and O'Malley et al. [51]).

Ensemble Learning for Text Generation

Furthermore, multiple learning algorithms could be used to obtain a better predictive performance than the one obtained from any of the constituent learning algorithms alone. [11] This approach is know as ensemble learning or ensemble methods. One example of such approach is Murugadoss et al. [48] work, who developed an automated de-identification system that employs an ensemble architecture, incorporating attention-based deep-learning models and rule-based methods. Ensemble methods can result in highly accurate models but evaluating the prediction typically requires more computation than evaluating the prediction of a single model. In many ways, ensemble learning is a way to compensate for poor learning algorithms by performing a lot of extra computation.[8] Their work performance performance on 2014 i2b2 de-identification dataset was better compared to other state-of-the-art models, but in the context of solving automatic de-identification together with automatic classification, there are simpler single models that could perform similarly with higher scalability and control. Additionally, their analysis of focused only on the performance of detecting PHI instances, meaning that the system fails to detect risk of re-identification based on semantics [48]. Li et al. [40] combine *Re-inforcement Learning* (RL),[12] GANs and RNNs to build a new model with high performance in the supervised task of *sentiment analysis*.

Conditional Text Generation (CTG)

As stated in the introduction, the effort in this work is into generating a novel or an surrogate version of a medical record, controlling the generative process

[9] method for propagating the total loss back into the neural network distributing it accordingly to what every node is responsible for.

[10] *Bilingual Evaluation Understudy* score, which indicates how similar is a text to another reference text, providing an overall assessment of model quality. A BLEU score of 1 means perfect similarity.

[11] based on `https://en.wikipedia.org/wiki/Ensemble_learning`

[12] RL problems concern learning what to do and how to map situations to actions inside an environment, so as to maximize a numerical reward signal. For instance, in online gaming.

by injecting codes with featural profiles to the latent representations. This is precisely the goal of CTG: to take condition-codes into consideration to influence the output during the generation process. These conditions usually include context, topic, emotion, external knowledge, and so on, but in theory any syntactic or semantic feature encoded in the text could be controlled. Biswal et al., 2020, [8] make use of a VAE-based model called EVA for synthesizing sequences of discrete Electronic Health Record (EHR) encounters (like clinical visits). Models like this and like the one from Hu et al. [27] perform well on short sentences and can be scaled to long text generation by the use of hierarchical structures, attention mechanisms, and extra latent space regularization techniques. Keskar et al. [31] released *CTRL*, a 1.63 billion-parameter conditional transformer language model, trained to condition on control codes that govern style, content, and task-specific behavior. They used a huge amount of training data and derived the control codes from the text structure preserving the advantages of unsupervised learning while providing more explicit control over text generation. A pre-trained model like this could easily escalate to several dataset of free text via transfer learning [13]. Other authors propose simpler approaches based on transformers, such as Dathathri et al. [14], who put forwards the *Plug and Play Language Model (PPLM)* for controllable language generation, which combines a pre-trained model with 1+ attribute classifiers that guide the generation without any further training of the language model. In general, attention based models capture long-term dependencies better than most of the alternatives, but training such models is costly and usually more complex due to the higher number of parameters and hyperparameters. Table 2.1[14] summarizes some pros and cons of the most popular text generation techniques.

The chosen model for this work is based on VAEs. Hu et al. [27] advanced a conditional language model with holistic attribute discriminators, which through collaborative learning, learns interpretable representations of text conditioned on a code *c*. For this purpose the authors used IMDB text corpus [15] and the *Stanford Sentiment Treebank-2* (SST-full) to train a generator and to sample from a latent space conditioned on the sentiment of the text (positive/negative). Their model is trained in two phases similarly to

[13]to take advantage of previously learned feature maps without having to start from scratch by training a large model on a large dataset. The process of refining the pre-trained model is called fine-tuning and is the essence of transfer learning.
`https://www.tensorflow.org/tutorials/images/transfer_learning`

[14]inspired by Guo et al. [24], whose work is focused on giving a comprehensive review of new research trends in CTG

[15]Movie Reviews dataset is a binary sentiment analysis dataset consisting of 50000 reviews from the Internet Movie Database `http://nlpprogress.com/english/sentiment_analysis.html`

Architecture	Advantages	Disadvantages
RNN	Natural structure for sequence modeling	Cannot capture long-term dependencies (context)
GAN	Its learning is unsupervised. Generate realistic samples	Unstable training. Cannot handle discrete data like text
VAE	It leverages the latent vector to increase diversity	Content is generated in spite of the quality
Transformers	Attention mechanism captures long-term dependencies	Very costly: long calculation times and slow training speed

Table 2.1: Common techniques for Text Generation: Advantages & Disadvantages

Dathathri et al. [14] work : one that trains the generator into creating plausible surrogate sentences, and a second one where the full model is trained alternating the optimization of the generator and the discriminator. They obtained meaningful generation results, but with restricted sentence length due to the chosen architectures for encoder and decoder and lack of regularization.

Hierarchical Structures and Regularization for Long-Text Generation

The length of the input and output text is a very important aspect for design decisions, when choosing an architecture for the blocks of the generative model. The VAE architecture has made and continues to make progress text generation tasks, but different research efforts have shown how encoder/decoder selections can translate into meaningful performance changes of the model. There is a particular relationship between each sentence; more specifically, between the latent variables that control the generation of the sentences. Relationships between these latent variables help in generating continuous and logically-connected long text. Zhao et al. [66] proposed a method for combining a *Transformer-Based Hierarchical Variational Auto-Encoder* (based on (BERT and GPT-2) and a *Hidden Markov Model* into a model (HT-HVAE) capable of learning multiple hierarchical latent variables and their relationships. A hierarchical VAE extends the basic idea by introducing a hierarchy of L latent variables and intermediate sentence representations into the generative networks to guide the word-level predictions. A hierarchically-structured[16] VAE alone already shows a significant improvement for generating long and coherent units of text (Shen et al. [56] and Shen et al. [57]). VAE and transformers-based models for CTG has also been proved useful for generating high quality and realistic electronic health sequences (Libbi et al. [41]), and gold standards

[16]loosely-coupled sub-nets arranged in layers (acyclic graph).

data augmentation systems in healthcare are of a lot of interest for clinical NLP researchers. However, as stated before, in this thesis work the author's efforts are into offering a simple but robust and scalable architecture for conditional text generation capable of solving the tasks of automatic de-identification and automatic classification of medical records, by the side of data augmentation and interpolation tasks. Implementations like HT-VAE from al., could be valuable extensions of this base architecture, the same way extra regularization techniques and design decisions for encoder/decoder are. The architecture proposed by Hu et al., 2018 [27], modified with extra regularization and to produce long coherent text seems suitable for the task and is the baseline for this thesis.

More methods to improve Text Quality

Moreover, to further alleviate the issue of noisy data and ensure robustness and uniform information density in the generated text by the learned language models, several techniques could be considered into being incorporated to the base model as extensions such as: using a minimum entropy regularization term, latent space enrichment methods like the ones presented by Li et al. [39], Li et al. [37] and Dieng et al. [16]), beam search (Meister et al. [44]) and bias correction (Grover et al. [22]). These efforts have shown that incorporating these methods is highly advantageously with potential general application to data scarce NLP learning tasks on healthcare, like i2b2 datasets. The details and effect of some of them are discussed later in this work.

Text Classification

Finally, from the side of automatic text classification, the participant teams submissions (reported by Uzuner et al. [61]), include supervised and unsupervised classifiers, and handcrafted rules based systems. Despite the differences in their approaches many of them produced good results. The best solutions came from systems that leveraged the existence or lack of explicit smoking status information on the records, meaning that they would most likely misperform inferring on a *smoke-blind*[17] dataset or inferring about another featural description like psychological status of the patient or his/her risk of develop a specific progressive disease. On the contrary and as stated before, deep learning based systems, don't make use of handcrafted rules and can mitigate the training dependency of the model, making it more general. The two main deep learning architectures for text classification are based on the RNN and

[17]data where all explicit mentions of smoking terms have been removed leaving only the nonsmoking-related text

the *Convolutional Neural Network* (CNN) architectures. As summarized in the work of Minaee et al. [46], CNNs work better where detecting local and position-invariant patterns is important. The CNN where firstly introduced into text classification by Kim [32] and performed remarkably well compare to state-of-the-art of the time, and is still widely used in NLP applications together with attention mechanism or as part of bigger language models, based for instance in conditional modelling. Hence, TextCNN is the chosen base architecture for the discriminator(s) blocks of the conditional model.

Chapter 3

Approach

3.1 Data Description, Pre-processing and Feature Engineering

In 2006, to increase the availability of clinical records and to contribute to the advancement of the state of the art in *Medical Language Processing* (MLP), within the i2b2 (*Informatics for Integrating Biology to the Bedside*) project , Uzuner et al. [60][61] de-identified and released a set of clinical records from the *Partners Healthcare Research Patient Data Registry*.[1][2] The records were initially pre-processed so that they were de-identified[3], tokenized, broken into sentences, converted into XML format, and separated into training and test sets. The free text in the original XML records (as in 2006's challenge) is structured by clinical titles such as the ones shown in Figure 3.1, and on the right image of Figure 3.2. A title is defined an upper-cased noun phrase followed by ':'. Details about the regular expression pattern used to match them can be found in the script *utilsEDA.py* inside the repository. There are around 2.000 different titles present on the free text of the dataset, outlining a different aspect of the patient clinical history.

In total, 1.391 records were released, providing the basis for the development of the ground truth for two clinical NLP challenges:

- Automatic de-identification of clinical data
- Automatic evaluation of the smoking status of patients based on medical records

[1] centralized clinical data registry. It gathers data from various hospital systems and stores it in one place. https://precisionmedicine.bwh.harvard.edu/resources/

[2] hosted on Harvard's Medical School DBMI Data Portal under the name of n2c2. https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/

[3] manually with the help of existing de-identification systems

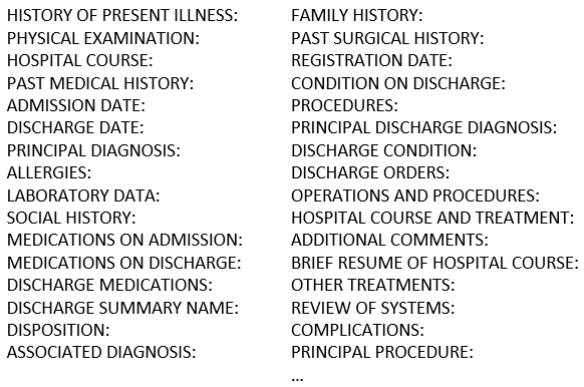

Figure 3.1: Preview of 32 unique *headers* present in the records.

3.1.1 Automatic De-Identification Challenge

A total of 889 medical discharge summaries were part of this challenge: 669 for training and 220 were used for testing. They were prepared by annotating and by replacing all authentic PHI with realistic surrogates. The datasets contain a total of 8 different PHI tags derived from the 18 categories defined by the HIPAA (Figure 1.2). Figure 3.2 shows an example of record from the training data in its original format. The original records for the de-identification challenge are annotated with PHI tags in the form $<$***PHI TYPE='_phi_type_'***$>$*_phi_*$<$***/PHI***$>$. The red rectangles inside this image refer to the subsections inside the records as described in Figure 3.1.

The goal of de-identification, as defined in the first challenge, was to find and remove PHI from medical records while protecting the integrity of the data[4], in the presence of *ambiguities*[5] and *out-of-vocabulary* PHI.[6] Ambiguous and out-of-vocabulary PHI reduce the contribution of dictionaries and gazetteers (NER approaches) to de-identification and emphasize the value of studying context and language (Uzuner et al. [60]).

PHI-Tags Categories

The 8 Private Health Information tags used in the original challenge were derived from the existing 18 categories defined by HIPAA. The PHI category

[4]retaining medical concepts, such as diseases, is important to enable the use of de-identified records for research[60].

[5]PHI and non-PHI can lexically overlap, e.g., Huntington can be the name of a disease (non-PHI) as well as the name of a person (PHI) [60].

[6]can include misspelled and/or foreign words that cannot be found in dictionaries[60].

Figure 3.2: Real training data samples. Left: de-identification challenge; Right: smoking status classification challenge.

'Names' is one of the most sensitive PHI inside the established HIPPA categories. So, here the author focuses initially in 3 of the PHI tags existing on the data: names of **patients**, **doctors** and **hospitals**, and transform the data according to the details about the pre-processing and binary re-labelling in section 3.3 of this chapter. The existence of any of this PHI tags in the text, define the *'Present'* or positive label in one of the output datasets, and are defined as follows [60]:

- **Patients:** includes the first and last names of patients, their health proxies, and family members. It excludes titles, such as 'Mrs.', e.g., 'Mrs. (Mary Joe) was admitted...'.
- **Doctors:** refers to medical doctors and other practitioners mentioned in the records. For transcribed records, it includes the transcribers' names and initials. It excludes titles, such as Dr. and MD, e.g., He met with Dr. [John Bland], MD.
- **Hospitals:** marks the names of medical organizations and of nursing homes where patients are treated and may also reside. It includes room numbers of patients, and buildings and floors related to doctors' affiliations, e.g., The patient was transferred to Gate 4.

3.1.2 Automatic Classification Challenge

A total of 502 medical discharge summaries were used for this challenge. These records contain non-tagged-PHI and were labeled with one of 5 smoking status categories by pulmonologists. The data is transformed to produce 2 experimental datasets. The 2 versions represent including or not the removal of headers from the text as a pre-preprocessing step as described in section 3.1.3.

Smoking Status Categories

For the purpose of the challenge, the smoking-status categories are defined as (Uzuner et al. [61]):

- **A Past Smoker** is a patient whose discharge summary asserts explicitly that the patient was a smoker one year or more ago but who has not smoked for at least one year.
- **A Current Smoker** is a patient whose discharge summary asserts explicitly that the patient was a smoker within the past year.
- **A Smoker** is a patient who is either a Current or a Past Smoker but whose medical record does not provide enough information to classify the patient as either.
- **A Non-Smoker** is a patient whose discharge summary indicates that they never smoked.
- **An Unknown** is a patient whose discharge summary does not mention anything about smoking. Indecision between Current Smoker and Past Smoker does not belong to this category.

3.1.3 Pre-processing and Binary Re-labelling

The recurrent appearance of the noun phrases that compose the titles in the free text, could lead the model to overfitting[7]. Part of the pre-processing steps include creating an dataset version removing this titles from the text.

For the de-identification challenge the training data was split by chunks of text of a defined length range. If chunk $\boldsymbol{x}$ contained PHI tags, then it was cleaned of them and labelled with the positive label: *Present*. The data for the smoking-status classification challenge was re-labelled similarly. Here, only

[7]the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably. https://en.wikipedia.org/wiki/Overfitting

the *Unknown* class represents that there is no information present about the smoking status of the patient in the text (negative class). Hence, the other 4 possible classes compose the positive label. Figure 3.3 illustrates this process. In result, there are 2 possible labels for each of the 2 possible attributes. However, the model can be easily adapted to include more labels and more attributes for multi-class and multi-label tasks.

- **Present**: There are signs/content of attribute i in the text. If attribute i is the smoking-status, then this implies that inferring the smoking-status is possible is the discriminator is trained to include all the categories as described in 3.1.2.
- **Not Present**: Attribute i cannot be found in text. If the attribute i is the PHI, this means that the generated record $\hat{x}$ is a de-identified version of x, which has been sampled from the latent representation z of the record .

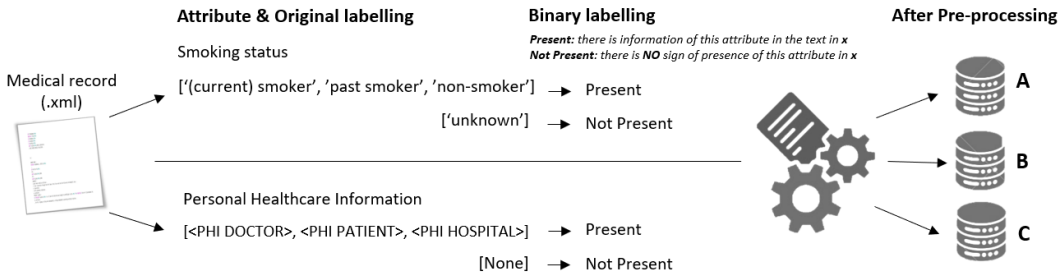

Figure 3.3: Generating A, B and C through binary re-labelling

Additional pre-processing steps include removing dates and most of non-textual information, and skipping some generated samples due to their short length.

Training Datasets

The three resultant datasets are:

- **Dataset A**: Unlabelled dataset composed of **669** of pre-processed medical records without PHI tags[8]. The average length of the records is 2439 words. A.1 represents the dataset as original using headers and A.2 is an alternative version where these titles have been removed.

[8]meaning that the removal of the XML format which indicates that the noun-phrase is of type PHI. For instance: <PHI TYPE="HOSPITAL"> Macomtoma Medical Center </PHI> → Macomtoma Medical Center

- **Dataset B:** Labelled dataset composed of **1824** chunked pre-processed medical with binary-labelled depending on the presence of PHI in the text. The average length of the records is 2439 words.
- **Dataset C:** Labelled dataset composed of **398** pre-processed medical records without titles nor PHI tags which has been binary-labelled depending on the presence of smoking-status information in the text. The average length of the records is 439 words.

The distribution of instances and tokens in the challenge corpus is shown in Table 3.1 (based on Uzuner et al. [60] report). This is an insight of the information content inside datasets A,B and C.

Type	Complete Corpus		Training Data		Test Data	
	Instance	**Token**	**Instance**	**Token**	**Instance**	**Token**
Non-PHI	-	444.127	-	310.504	-	133.623
Patients	929	1.737	684	1335	245	402
Doctors	3.751	7.697	2.681	5.600	1.070	2.097
Hospitals	2.400	5.204	1.724	3.602	676	1602

Table 3.1: Instance & token distribution: de-identification challenge data.

3.2 Architecture[9]

Variational Auto-encoder

Variational Auto-encoders receive this name because their remind AEs, but in reality they work in a very different way. The AE is a self-supervised technique and one of the most powerful ideas in deep learning architectures for *Latent Representation Learning*. In simple words, *auto-encoding* is a data compression algorithm where the compression and decompression functions are learned automatically (through the neural network weights for encoder and decoder). AEs are in general too data-specific[10] and lossy.[11] These facts make them impractical for real-world data compression problems but still useful as a dimensional reduction and data denoising method (Rocca [54]). As a generative

[9] global structure you want to build your neural network on. For instance connecting some convolutional layers to extract features, some fully connected layers, and finally a softmax layer to make predictions.

[10] fit to be able to compress data similar to what they have been trained on

[11] similar to JPEG compression, the outputs will be degraded compared to the original inputs.

model they fail as well, due to the irregularity of its latent space representation, which depends on the distribution of the data in the initial space, the dimension of the latent space and the architecture of the encoder. Because of the fact that an AE is trained to encode and decode with as few loss as possible, they really don't care about how the latent space is organised and cannot ensure an smart way of doing it, leading them to overfitting. The problem is that this also implies that some points of the latent space will give meaningless content once decoded as shown in A in Figure 3.4[12]. Regularization creates a *gradient* over the information encoded in the latent space. Therefore, a point which is halfway between the means $\mu_z(x)$ of the encoded distributions coming from different inputs (color clouds), should be decoded in something that is somewhere between the data points that gave the initial (reflected by shape and color distributions).

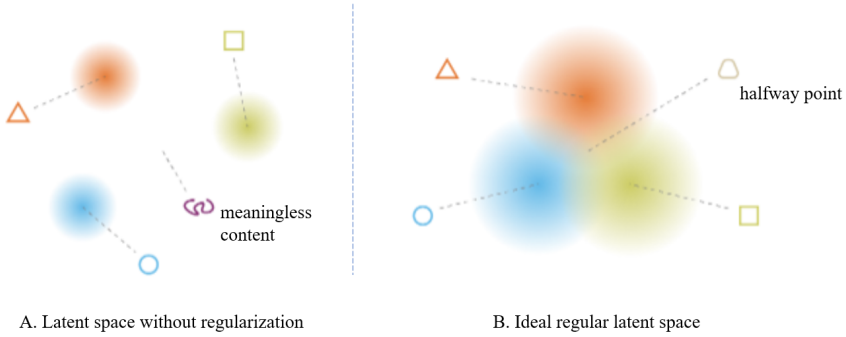

Figure 3.4: Effect of regularization on the latent space .

On account of involving some regularization, instead of encoding an input as a single point x, the Variational Auto-encoder encode it as a distribution $p(z|x)$ over the latent space. Then, it samples points from that distribution in the latent space, decodes them and calculates the error to back-propagate it through the network. The encoder model is also referred to as the *recognition model* while the decoder model as the *generative model*. Now, the amount of information is that is lost is measured using the *reconstruction log-likelihood* $\log p_\phi(x \mid z)$. This measure tells us how effectively the decoder has learned to reconstruct an input x given its latent representation z, which is inferred by a conditional probabilistic encoder E. Figure 3.5 illustrates in high level the encoding and decoding process. The latent variable or code z is then:

$$z \sim E(x) = p(z|x) = q(z|x) \tag{3.1}$$

[12]modified image from Rocca [54]. Used here with the permission of the author.

The statistical motivation of the VAE originates from variational inference principles, with the aim of providing a approximate solution $q(z|x)$ to an intractable distribution $p(x)$. In simple words, If starting by an observation x which is generated by some hidden variable z, this we would like to infer the characteristics of z, this is $p(z|x)$:

$$p(z|x) = \frac{p(z|x)p(z)}{p(x)} \tag{3.2}$$

But the integral for computing $p(x)$ is computationally intractable:

$$p(x) = \int p(x|z)p(z), dz \tag{3.3}$$

In essence, it is possible to define the parameters for an tractable distribution $q(z|x)$, such that is similar to the intractable $p(z|x)$, and perform inference with it. In other to compare the distributions, the *Kullback-Leibler Divergence* is used. This function measures how different two different distributions are,[13]. This divergence measures how much information is lost when using q to represent p, and can be understood as a measure of how close is q to p. Is calculated as:

$$KL(q||p) = \sum_x q(x) log \frac{q(x)}{p(x)} \tag{3.4}$$

The fact that VAEs encode inputs as distributions instead of simple points is not sufficient to ensure continuity[14] and completeness.[15]. As stated by Rocca [54], without a well defined regularisation term, the model can learn to minimise its reconstruction error by ignoring the fact that distributions are returned, behaving almost like classic autoencoders, and leading the model to overfitting.[16] In order to avoid these effects, we have to regularise both the co-variance matrix and the mean of the distributions returned by the encoder, pushing them into being close to the identity and close to 0 respectively. This is the reason why the total loss of the variational autoencoder for a single data point is composed by the reconstruction loss (negative log-likelihood [17])

[13] $KL(p||p) = 0$; KL-divergence is 0 for two identical distributions.

[14] two close points in the latent space should not give two completely different contents once decoded [54]

[15] for a chosen distribution, a point sampled from the latent space should give "meaningful" content once decoded (Rocca [54])

[16] the encoder can either return distributions with tiny variances (punctual distributions) or return distributions with very different means (far apart from each other in the latent space) (Rocca [54]).

[17] cost function in ML, that tells how bad the model it's performing; the lower the better. urlhttps://medium.com/deeplearningmadeeasy/negative-log-likelihood-6bd79b55d8b6

and the *KL-divergence* between $q(z|x)$ and $p(z)$ as a regularization. The full derivation of the loss function begins from minimizing $KL(q(z|x)||p(z|x))$ and it's explained in detail in works such as the one from Doersch [17] , Asperti and Trentin [6], and Kuleshov and Ermon [34].

$$loss_{vae} = log(p(x)) - KL(q(z|x)||P(z|x)) \tag{3.5}$$

$$loss_{vae} = E_{q(z|x)}logp(x|z) - KL(q(z|x)||p(z)) \tag{3.6}$$

The *KL-divergence* is always positive, hence the maximization of equation 3.5 is precisely the learning objective of VAEs and is the so known *Evidence Lower Bound* or *ELBO* (Asperti and Trentin [6]). A simpler way of writing the reconstruction loss instead as the negative log-likelihood between distributions is by expressing it as the *Squared Euclidean Distance*[18] between and input x and the output of the decoder $d(z)$:

$$E_{q(z|x)}logp(x|z) => ||x - d(z)||^2 = ||x - \hat{x}||^2 \tag{3.7}$$

In VAEs, $p(z)$ is the enforced known prior distribution, and is specified as a *Standard Normal Distribution* with zero mean and variance equal to one ($p(z) = Normal(0, 1)$). In other words, VAEs assume that there is no simple interpretation of the multiple dimensions of z, and instead claim that samples of z can be drawn from a simple distribution, the normal distribution $N(0, I)$. If the encoder outputs representations z that are different than those from a standard normal distribution, it will receive a penalty in the loss, keeping the representations z of each digit sufficiently diverse. Figure 3.5[19] illustrates the variational encoding and decoding process.

In this order of ideas, a VAE computes, for each latent variable z and each sample x, an expected value $\mu_z(x)$ and a variance $\sigma(x)$ around it. During training, the variance generally drops very fast to values close to 0, reflecting the fact that the network is highly confident in its choice of $\mu_z(x)$. The *KL-divergence* in the loss function can be also understood as a term aimed to reduce this confidence, by forcing a non-negligible variance. Considering this, the loss function of the VAE can be rewritten as:

$$loss_{vae} = ||x - \hat{x}||^2 - KL(N(\mu_x, \sigma_x), N(0, 1)) \tag{3.8}$$

As for any regularisation term, this comes at the price of a increase of the reconstruction error on the training data. As proved by Asperti and Trentin [6] in their work, the former will try to improve the quality of the reconstruction,

[18] sometimes referred simply as quadratic distance in the literature [6]

[19] modified image from Rocca [54]. Used here with the permission of the author.

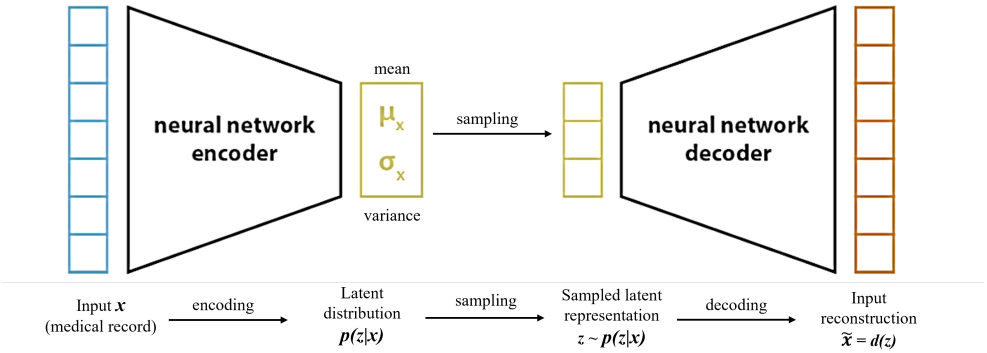

Figure 3.5: VAE diagram + high level data flow from real x to surrogate $\hat{x}$.

neglecting the shape of the latent space; on the other side, *KL-divergence* is normalizing and smoothing the latent space, possibly at the cost of some additional "overlapping" between latent variables, eventually resulting in a more noisy encoding. The trade-off between the reconstruction error and the *KL divergence* can however be adjusted by a parameter β. A VAE that takes this into consideration is sometimes called βVAE in the literature. The total loss can be then rewritten as:

$$loss_{vae} = ||x - \hat{x}||^2 - \beta \cdot KL(N(\mu_x, \sigma_x), N(0, 1)) \tag{3.9}$$

Now, there is one more point to consider. The sampling process has to be expressed in a way that allows the error to be back-propagated through the network, and gradient descent is not possible for a random sampling process. To alleviate this, a simple transformation knows as *re-parametrization*, uses the fact that if z is a random variable following a Gaussian distribution with mean μ and with covariance σ then it can be expressed as:

$$z = \mu + \sigma \odot \epsilon \tag{3.10}$$

Epsilon $\epsilon \sim Normal(0, 1)$, meaning that the idea is to randomly sample ϵ from a unit Gaussian, and then shift the latent distribution's mean μ and scale it by the co-variance σ. Using this re-parametrization, the parameters of the distribution can be optimized while still maintaining the ability to randomly sample from that it. Figure 3.6 (Jordan [30]) illustrates the described transformation:

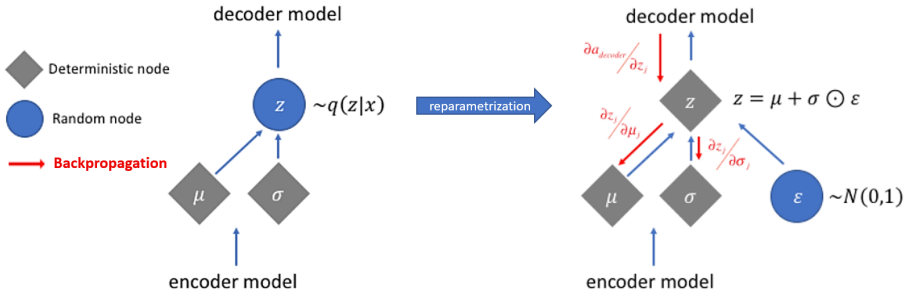

Figure 3.6: Re-parametrization transformation for latent z.

Wake-sleep Mechanism

This algorithm proposed by Hinton et al. [25] for learning deep generative models like *Helmholtz Machines* (Dayan et al. [15]), consists of two phases: wake and sleep, which optimize the generative model and inference model respectively. The updates in the wake phase updates the generator parameters θ by fitting $p_\theta(x|z)$ to the real data and hidden code inferred by the inference model $q_\lambda(z|x)$. On the other hand, the sleep phase updates the parameters λ based on the generated samples from the generator. VAEs can be seen as extending the wake phase by also optimizing the inference model q, with additional prior regularization on code z. Additional details about the relationship. The connection between VAEs, GANs and the wake-sleep algorithm is formally explained in the work of Hu et al. [26].

Encoder and Decoder design: The Gated Recurrent Unit (GRU)

RNNs suffer from a short-term memory problem. For long text like the medical records, they are not capable of carrying information from earlier time steps to later ones. Meaning that RNNs may forget important information from the beginning of the text. From the training (back-propagation) perspective, they suffer from the *vanishing gradient* problem. Gradients are values used to update a neural networks weights. The vanishing gradient problem is when the gradient shrinks as it back propagates through time. If a gradient value becomes extremely small, it does not contribute too much learning, preventing the network to learn fit representations (Phi [52]). . Almost all state of the art results based on RNNs are achieved with these two networks. Both *Long Short Term Memory networks* (LSTMs) and *Gated Recurrent Units* (GRUs) make use of internal mechanisms called gates that can regulate the flow of information, solving the short-term memory problem. Figure 3.7 (Gunawan et al. [23]) shows the gates inside the LSTM and GRU blocks.

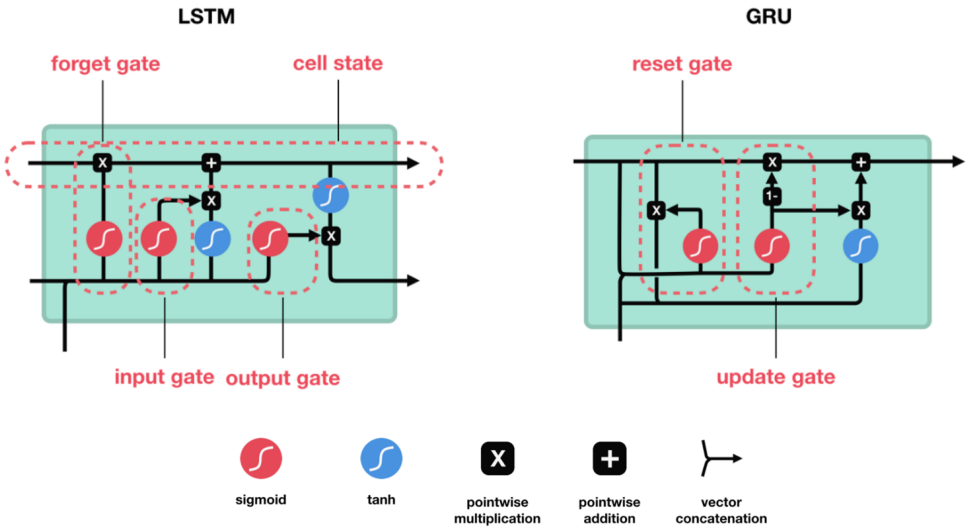

Figure 3.7: LSTM and GRU cells with inside gates.

Technically GRUs are similar to LSTMs but GRUs use fewer parameters and only two gates: the update and reset gates. The update gate tunes the update speed of the hidden state while the reset gate decides how much of the past information to forget by resetting parts of the memory (Aggarwal [3]). These gates can learn which data in a sequence is important to keep or throw away. By doing that, it can pass relevant information down the long chain of sequences to make predictions (Phi [52]). GRUs are defined by a set of 5 equations but in practice these units are easily constructed using PyTorch library.[20] Thereafter this mathematical description is negligible information for the scope of this work.

As reported by Cahuantzi et al. [13], an increase of RNN depth does not necessarily result in better memorization capability when the training time is constrained; when not, then LSTMs are better for more complex input text. On the side of that, their results also indicate that the learning rate and the number of units per layer are among the most important hyper-parameters to be tuned (Cahuantzi et al. [13]). GRUs are the chosen base architecture for encoder and decoder of the VAE, because they deal well with long-memory dependencies and are simpler and easier to train than the LSTM. Nevertheless, these decisions can be optimized in an ensemble learning setup so the chosen design is the most suitable for a specific application.

[20] machine learning framework that enables fast, flexible experimentation and efficient production. `https://pytorch.org/features/`

Convolutional Neural Networks for Text Classification (TextCNNs)

Kim [32] reported on a series of experiments with CNNs trained on top of pre-trained word embeddings[21] for sentence classification tasks, achieving great results in multiple NLP tasks such as *Sentiment Analysis* and *Question Classification*. TextCNN architecture and its transformations from input to output, are shown in Figure 3.8[22], and is the base chosen architecture to prototype the discriminator in the conditional text generation setting:

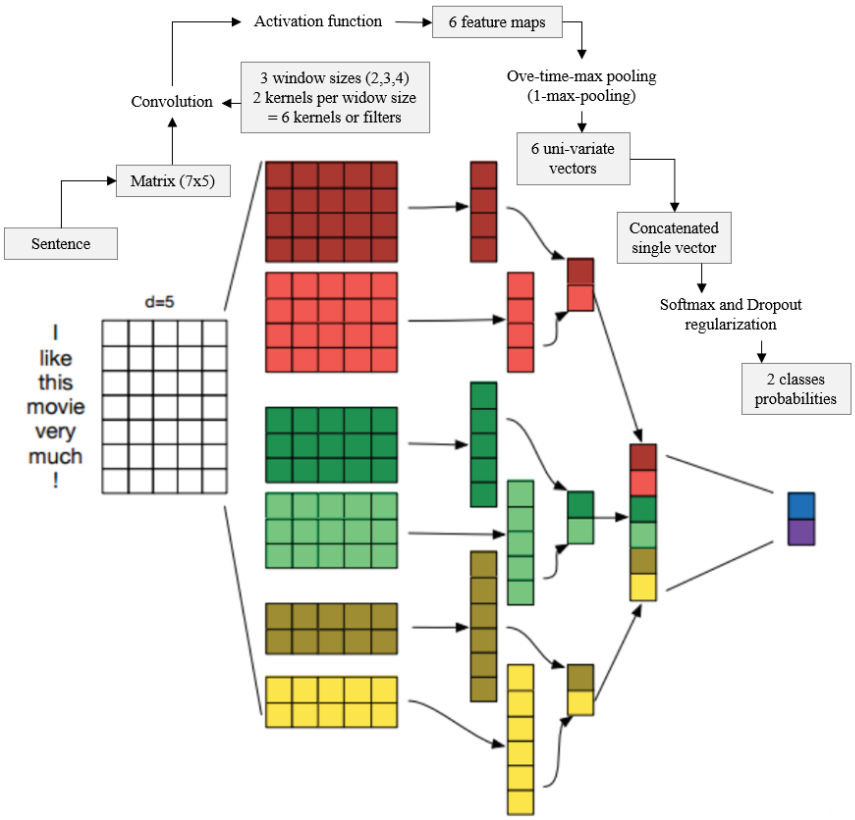

Figure 3.8: TextCNN sentence transformations to prediction.

Firstly, a sentence of length n is represented as:

$$x_{1:n} = x_1 \oplus x_2 \oplus ... \oplus x_n \tag{3.11}$$

where $x_{i:n}$ or $x_{i:i+j}$ refers to the concatenation of words and $\oplus$ is the concatenation operator. A convolution operation involves kernels or filters w_c,

[21] vectors of n dimensionality which are an efficient dense representation of words; hence, similar words have a similar encoding.
`https://www.tensorflow.org/text/guide/word_embeddings`

[22] modified image from Le et al. [36], posted on CoRR journal.

which is used to extract the features. Mathematically, they are matrices that move over the input data (a window of words), perform the dot product with the sub-region of input data, and gets the output as matrices of dot products. These kernels are applied to each possible window of h words in the sentence to produce *feature maps*:

$$c = [c_1, c_2, ..., c_{n-h+1}], \tag{3.12}$$

where each feature c_i is the activation—via a non-linear function f like the hyperbolic tangent— of the convolution of the kernel w_c with a window of h words, plus a bias term b :

$$c_i = f(w_c \cdot x_{i:i+h-1} + b) \tag{3.13}$$

The convolution operation for text is very similar to one with images, but in this case it's applied to 1-dimensional vector of words. Subsequently, the most important feature (highest value) for each feature map is capture through a *max-over-time pooling* operations.

$$\hat{c} = max(c) \in R \tag{3.14}$$

The difference with *max-pooling* lies in the fact sentences naturally have different length in a corpus, making the feature maps different for different sentences, and what is wanted is to reduce the tensor[23] to a fixed size so it's possible to apply softmax[24]. The resulting filters are concatenated to form a shallow-and-wide network as described by Le et al. [36]:

$$g = [\hat{c_1}, \hat{c_2}, ..., \hat{c_m}] \tag{3.15}$$

where m is the total number of filters. The network end with the class predictions thanks to a fully-connected layer as described—without dropout regularization[25]—in equation 3.16:

$$\hat{y} = f(w_y \cdot g + b_y) \tag{3.16}$$

As detailed in Section 3.1, we transcribe the automatic de-identification and smoking status challenges, into two binary classification tasks for a couple of TextCNN discriminators D_1 and D_2, part of a conditional generative model as described further in this chapter.

[23]array of numbers arranged on a regular grid with a variable number of axes [21].

[24]mathematical function that converts a vector of numbers into a vector of probabilities, proportional to the relative scale of each value in the vector. Each value in the output of the softmax function is interpreted as the probability of membership for each class [12]

[25]to not consider neurons during the training phase of certain set chosen at random

Controlled Text Generation: CVAE

The idea is to come up with a multitask framework for clinical NLP with scarce data, competent resolving automatic de-identification and classification of medical records among other applications. Thus, what is needed is a model that instead of penalizing the error between a single prediction and the ground truth, is capable of performing controlled generative tasks, such as mapping a single record with personal information to many possible outputs where the personal content has been removed/obfuscated; and discriminative tasks, like inferring the smoking status of a patient from the information present in the medical record. In the context of generative modelling, this is an algorithm that takes in text and produces a complex, multi-modal distribution that we can sample from. The *Conditional Variational Auto-encoder* (CVAE) allow us to tackle problems where the input-to-output mapping is one-to-many, without requiring us to explicitly specify the structure of the output distribution (Doersch [17]).

Essentially, in a *Conditional Generative Model* (CGM), for a given observation x, z is drawn from the prior distribution $p(z|x)$, and the output y is generated from the distribution $p(y|x, z)$. The latent variables z allow for modeling multiple modes in conditional distribution of output variables y given input x, making CGM capable of for modeling one-to-many mapping (Sohn et al. [59]).

The CVAE is composed of multiple *Multi Layer Perceptrons*,[26] such as the *recognition network* $q(z|x, y)$, the *(conditional) prior network* $p(z|x)$, and the *generation network* $p(y|x, z)$; and need to compute three functions: the decoder distribution $\mu(x, z)$, and $\hat{\mu}, \hat{\sigma}$ which comprise the encoder distribution $q(z|x, x)$ [62].

As explained by Hu et al. [27] in their work, the idea is to augment the unstructured variables z with a set of structured variables c, in order to control the attributes of interest in an interpretable way. The generator should the be conditioned on the combined vector $z|c$, and generate samples that fulfill the attributes as specified in the structured code c.The target attributes for c are PHI and smoking status information presence in the records as described in section 3.1, Figure 3.3, but in theory it could control any salient and independent syntactic or semantic feature of the medical records.

These models are trained to maximize the *conditional log-likelihood*, predicting the original medical record. The objective function for VAE as seen in equation 3.6, is now written for a CVAE as:

[26] *artificial neural networks* (ANNs) composed of multiple layers of perceptrons (with threshold activation)

$$loss_{cvae} = E_{q_E(z|x)q_D(c|x)} log p_G(x|z,c) - KL(q_E(z|x)||p(z)) \tag{3.17}$$

where E, D and G correspond to Encoder, Discriminator and Generator respectively. A more detailed description of this model and how it's trained can be be found in the work of Hu et al. [27] and in the next section.

3.3 Model and Pipeline: CVAE-TextCNN

In the VAE proposed by Hu et al. [27], a generator is trained to reconstruct real sentences in an "extended" wake-sleep procedure, and is combined with set of independent[27] attribute discriminators for imposition of structures c to augment the unstructured latent code z. The "extended" wake-sleep procedure refers to having a wake phase which updates the generator with samples generated from the inference network on training data, while the sleep phase updates the inference network based on samples from the generator, enabling collaborative learning with little supervision.

The model's structure is shown in Figure 3.9[28] . In general terms, as described by Hu et al. [27], an encoder E takes an input x and produces a latent vector z . Thereafter, a structured controllable vector c is defined, concatenated with the unstructured latent code z, and fed to the generator G to generate the corresponding sentence $\hat{x}$. The discriminator(s) D_i ensures that the generated sentence is consistent with the controllable vector c.

In order to prevent the potential dependence of the structured code with attributes not explicitly encoded, Hu et al. [27] introduced an independence constraint by training the generator so that other non-explicit attributes can be correctly recognized from the generated samples and match the unstructured code z. For this task, the variational encoder E is re-used and trained by minimizing the L_{vae} in equation 3.6. With this in mind, the generator's G optimization objective is defined by:

$$L_G = L_{vae} + \lambda_c L_{Attr,c} + \lambda_z L_{Attr,z} \tag{3.18}$$

where the first term corresponds to maximizing the likelihood of predicting the original medical record x given the latent spaces and the generator. $G(z,c)$, $L_{Attr,c}$ (equation 3.19) corresponds to maximizing the likelihood of generating the output documents $\hat{x}$ with the desired injected structured code

[27] imposing different attribute codes will keep the unstructured attributes invariant as long as z is unchanged

[28] based on Hu et al. [27]

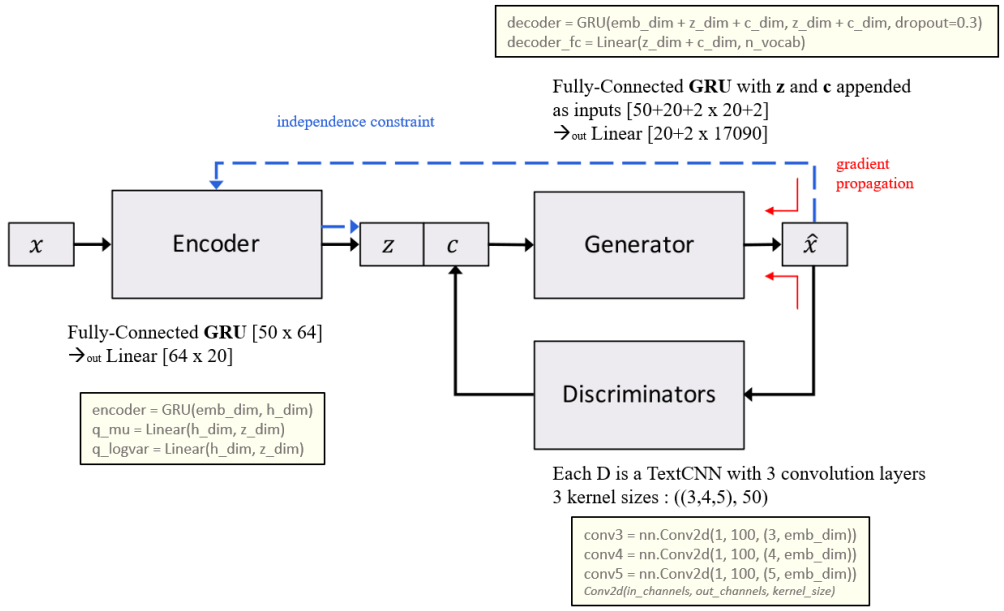

Figure 3.9: Pipeline with networks details and hidden connections

c (style entanglement), and $L_{Attr,z}$ (equation 3.20) refers to the loss due to the independence constraint, for which the encoder E is re-used to regenerate the latent distribution z devoid of the structured code c , from the output distribution $\hat{G}_\tau(z, c)$ [27]. λ_c and λ_z are are balancing parameter between the optimization objectives, the same way β is for the reconstruction loss and the KL-divergence in equation 3.9. $L_{Attr,c}$ and $L_{Attr,z}$ are calculated as follows:

$$L_{Attr,c} == -E_{p(z)p(c)} log q_D(c|\hat{G}_\tau(z, c)) \tag{3.19}$$

$$L_{Attr,z} = -E_{p(z)p(c)} log q_E(z|\hat{G}_\tau(z, c)) \tag{3.20}$$

with $q_D(c|x)$ as the conditional distribution—given an observation x—defined by the discriminator D for each structured variable in c.

$$D(x) = q_D(c|x) \tag{3.21}$$

$$z \sim E(x) = q_E(z|x) \tag{3.22}$$

and τ corresponds to an annealing temperature term initialized in 1 and pushed towards 0 as the training proceeds. The need for this arise—again, as explained by Hu et al. [27]—from the difficulty of applying the discriminators as the model proposed, as text samples are discrete and non-differentiable,

breaking down gradient propagation from the discriminators to the generator. The use of this decreasing temperature, introduces a continuous approximation based on softmax which anneals to the discrete case during training. The foundations for the use of this method (based on *Simulated Annealing* search) are well explained in the work of Li et al. [38]. Thereafter, a generated token[29] $\hat{x}_t$ at step t involves discrete decision making. The token is sampled from a multinomial distribution parametrized using softmax function at each time step t, which inputs a scaled $\mathbf{o}_t$ (logit vector[30]) as follows:

$$\hat{x}_t \sim softmax(\frac{\mathbf{o}_t}{\tau}), \tag{3.23}$$

Now, the code c represents in his work, a binary decision on whether to include or not information about two aspect of the patient: personal health information and smoking status information as described in section 3.1, Figure 3.3. For instance, if the controlled aspect is the smoking status information presence in the surrogate medical record, then c is one of $[0, 1]$. A decision of $c = 0$ for an input medical record x with ground truth "current smoker", would imply that the output surrogate $\hat{x}$ is purged of this attribute, meaning that ideally any syntactic or semantic sign of the smoking status of the patient would be gone, and the new label—given by fitted discriminator D— would be *Unknown*. In a similar way if the controlled attribute is the personal health information, the conditional model should be able to de-identify the medical record producing a surrogate that is trained to look like and keep the essence of x, without exposing private information from the patient as the original record was. To *keep the essence* means that as a minimum if the original record x was labelled as *Present* for the attribute A, and then conditioned on purging attribute B from it, the record should still be classifies as *Present* for attribute A, but *Not Present* for B. These 2 discriminators are trained using separate labeled data for the respective attribute as described in section 3.1. This is one huge advantage of this model, because it also mean that it is easy to arbitrarily combine a set of discriminators controlling the targeted attributes.

Figure 3.10 shows the unrolled pipeline with the 3 back-propagation signals that compose the optimization objective of the generator as show in equation 3.18:

As for the discriminator(s), it learns in collaboration with the generator to infer about the sentence attributes and to evaluate the error of recovering the desired feature as specified in the latent code c. For this case with categorical attributes (discrete), the discriminator can be formulated as a binary sentence

[29] instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing [43]

[30] vector of raw predictions that a classification model generates

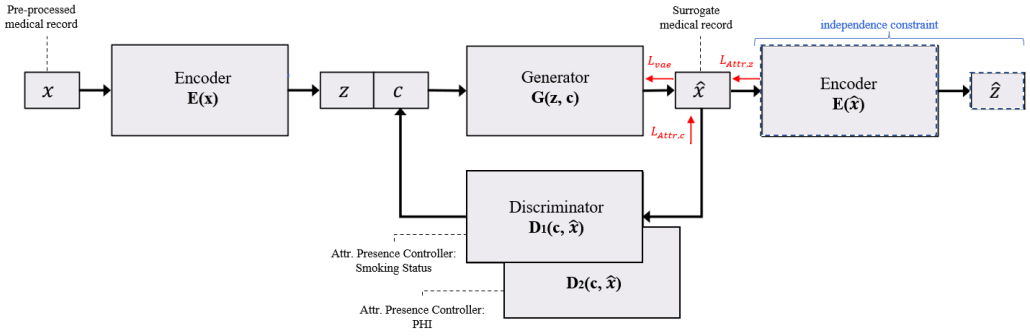

Figure 3.10: Unrolled model pipeline.

classifier. The structured variable c is learned using labelled data, as opposed to the unstructured code z which is learned in an unsupervised manner. To learn specified semantic meaning, a set of real labeled examples $X_L = (x_L, c_L)$ is used; nevertheless, the conditional generator G is also capable of augment the training data by synthesizing text-attribute pairs $(\hat{x}, c)$ (semi-supervised learning). To alleviate the noise in the data and ensure robustness, a minimum entropy regularization term is also introduced as part of the total training objective of the discriminator. The join objective is then:

$$L_D = L_s + \lambda_u L_u \tag{3.24}$$

where λ_u is a balancing term, L_s is the discriminator objective for the classification of labelled samples, calculated as:

$$L_s = -E_{X_L}[logq_D(c_L|x_L)] \tag{3.25}$$

and L_u is the minimum entropy regularization term due to classification of surrogate medical records $\hat{x}$, which drives the model into having high confidence in its predictions:

$$L_u = -E_{pG(\hat{x}|z,c)p(z)p(c)}[logq_D(c|\hat{x}) + \beta H(q_D(c'|\hat{x}))] \tag{3.26}$$

where $H(q_D(c'|\hat{x}))$ is the *Empirical Shannon Entropy* of q_D evaluated on the generated sentence $\hat{x}$; and β is another balancing parameter. It quantifies the amount of uncertainty for an event measured in bits, and provides a measure of the average amount of information needed to represent an event drawn from a probability distribution for a random variable (Brownlee [11]).

3.4 Training Algorithm

The training of the conditional model is done in two phases, one that updates the weights of the generator, and one that updates the weights of the discriminator, resembling the wake-sleep procedure with some extra details. In this context— additionally to the learning algorithm described in Figure 3.11—the model works in 2 modes in which [27]:

Sleep-phase (extended): samples are produced by the generator and used as targets for maximum likelihood training of the discriminator. Additionally, the generated samples are leveraged to improve the generator (*dream* samples obtained through *Ancestral Sampling* from the generative network). *Ancestral sampling*, as defined by Bishop et al. [7] in his book, is "the process of producing samples from a probabilistic model by first sampling variables which have no parents using their prior distributions, then sampling their child variables conditioned on these sampled values, then sampling the children's child variables similarly and so on...".

Wake-phase: samples c from the discriminator distribution $q_D(c|x)$ on observation x, to form a target for training the generator.

The work of Hu et al. [27] is based on short sentences x and sentiment classifiers D_i. As an addition, the author here argues to adapt the model for long text and equally obtain a stable discrete latent code with holistic discriminator metrics for arbitrary attribute of medical records.

Algorithm: Text Generation Controlled by D_i

Inputs:

- Unlabeled text corpus $X = \{x\}$
- Labeled corpus $X_L = \{(x_L, c_L)\}$
- Value for balaning parameters λ_c, λ_z, λ_u, β

Steps:

1. Initialize the VAE by minimizing L_{vae} on X, with c sampled from prior $p(c)$

repeat

2. Train the discriminator D_i by minimizing L_u

3. Train the generator G and the encoder E by minimizing L_G and L_{vae}

until convergence

Output: trained text generator G conditioned on disentangled representation (z, c)

Figure 3.11: Conditional Text Generation algorithm

3.5 Implementation

The model used in this work was based on the worked of Hu et al. [27] as stated before. Some parameter values were tuned to the challenge purpose

and for long text generation, and the pre-processing steps were design for the datasets of the challenges; simple manipulations were imposed like extra regularization, gradient monitoring and *gradient clipping*, and GRU units for encoder/decoder instead of RNN or LSTM. The code is written in Python, adapted from the base provided by the University of Bonn's NLP Lab project on Winter Semester 2017/2018.[31].

The best version of the models which are described in the next chapter, were trained using *Gammaweb Cluster*[32] from Bauhaus University Weimar, for 3.000 iterations in the first phase (generate pre-trained VAE) and 100.000 iterations in the second. To accomplish this, the application was *dockerized*,[33][34] uploaded to the *Docker Hub* and downloaded into *Betaweb Cluster* for posterior execution. Figure 3.12[35] illustrates the process together with the most important python scripts inside the container.

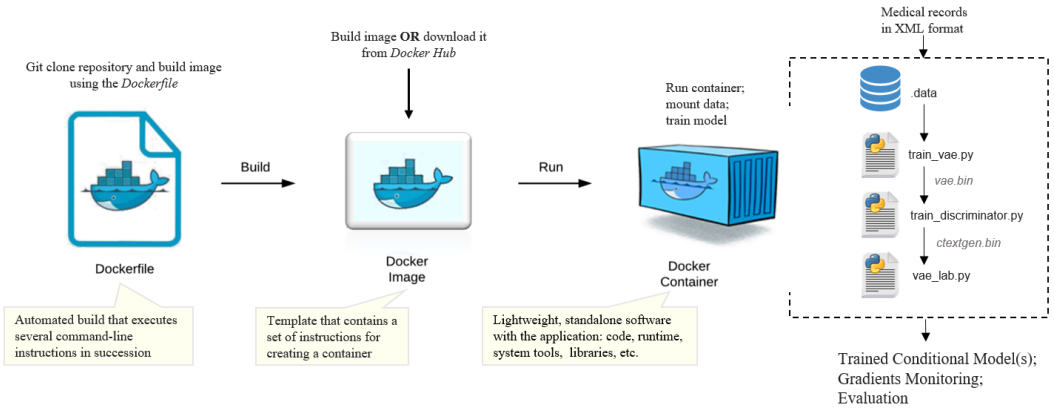

Figure 3.12: Dockerization and important files.

The steps for correct execution and additional information about the structure of the project can be found inside the repository for this work hosted by Bauhaus University Weimar.

[31] base code: https://github.com/wiseodd/controlled-text-generation

[32] with 9 nodes, 360 cores at 2.1 GHz, 7.5 TB RAM, 70.5 TB SSD, 24 Nvidia A100, 24 GeForce GTX 1080

[33] Docker is an open source tool that ships an application with all the necessary functionalities as one package.

[34] dockerizing means packing + deploying + running applications using Docker containers.

[35] the icons were taken from *Implementing Embedded Continuous Integration with Jenkins and Docker*
`https://community.arm.com/arm-community-blogs/b/tools-software-ides-blog/posts/`

Chapter 4

Evaluation

4.1 Experiments

Goal: Starting from a pre-trained language model (word embeddings) and a vanilla VAE, train a generator to reconstruct realistic surrogate medical records $\hat{x}$, conditioned on the latent code $z|c$; and a discriminator to binary-classify the records depending on the presence of attribute i in the text. c is binary-conditioning the latent z to impose semantic structures for 2 attributes.

$$c(x, Attr_i) = \begin{cases} 1 & \text{if } Attr_i \text{ is } present \text{ in text } x \\ 0 & \text{if } Attr_i \text{ is } NOT \text{ } present \text{ in text } x. \end{cases} \tag{4.1}$$

with $Attr_i$: *[PHI, Smoking Status Information]*. Each of the discriminators D_i corresponds to controlling one attribute:

- D_1: smoking status presence classifier. In the context of the original 5 categories described in chapter 3.1, this is a smoking-status binary-classifier where 0 corresponds to the *Unknown* category and 1 corresponds to any of the other 4.
- D_2: PHI presence classifier (for any of the 3 PHI types: *Doctor, Patient, Hospital*).

Each of the experiments was repeated for several settings changing the number of epochs[1] in the wake and sleep phases and applying hyperparameter tuning. The training procedure was stopped when convergence of the KL-term or reconstruction loss in the optimization objective equation. The stored

[1] training iterations

models include only the best versions for each experiment, and the number of epochs is reported in Section 4.3.

Pre-trained word embeddings are used to leverage learned representations from large datasets and transfer the knowledge to other data and NLP tasks. They are specially useful for data-scarce NLP applications where generalizability is relevant. The embedding weights used for this work are pre-trained GloVe Vectors[2] trained on *Wikipedia 2014* and *Gigaword 5*[3] datasets to produce 6 billion token-representations. The embeddings from words that are part of the training data, are still updated during the two phases of the training procedure.

4.1.1 I. Controlling the smoking-status presence

Here, the author propose to train a vanilla VAE on Dataset A, creating a regular latent z, to which semantic structures c for smoking-status information will be imposed. Due to the Independence constraint, if the original record contained PHI, then it should be classified as such when inferring with pre-trained discriminator D_2 for PHI presence (output of experiment II). After the model has converged, trained D_1 will be able to solve the task for the automatic smoking-status classification challenge, with the difference that here the discriminator only binary decide whether or not the text x contain enough information for it to be labelled as any of the first 4 smoking status categories, or if the label is *Unknown*, as described at the beginning of chapter 3.1. In theory the TextCNN architecture could be trained alone to solve the challenge without considering conditional generation models of any kind, yet the author consider the experiment to test the quality of the model in another setting. The generated record does not have to be particularly driven to have a content structure of any kind, this means that the balancing parameter in equation 3.9 can be ignored or set to a fixed neutral value.

For this purpose we define the variables according to the training algorithm in Figure 3.11 as:

- $D_i = D_1$
- Unlabelled text corpora X = Dataset A.1 and A.2 = x
- Labelled text corpus X = Dataset C = x
- c refers to controlling the content of semantic structures of PHI in the latent code, to generate an $\hat{x}$ that is exposing or not sensible information

[2]Global Vectors for Word Representation: https://nlp.stanford.edu/projects/glove/
[3]newswire text data : https://catalog.ldc.upenn.edu/LDC2011T07

of the patient's record, such as his/her name, hospitals names or any doctors names.

After the first phase is done by minimizing L_vae, the model outputs a generator capable of sampling surrogate medical records from the unstructured latent z through $E(x)$. For this experiment the first phase is doubled, this means that after the doubled extended sleep phase we have two generators trained with unsupervised A.1 and A.2 respectively. After the second phase (wake) has converged, the model outputs are:

- Trained generators G_{I_1} and G_{I_2} conditioned on disentangled representation (z, c) capable of reconstructing an $\hat{x}$, performing data augmentation controlling the amount of smoking-status content, and performing interpolation between points in that latent $(z|c)$.
- Trained discriminators D_{11} and D_{12} capable of deciding whether or not text x contains smoking-status information.

These two blocks are comprised in a pair of single models *ctextgen_I1* and *ctextgen_I2* that represent the conditioned generative models as explain in chapter 3.1, section . However, the stability of *ctextgen_I2* was severely compromised after the second phase of the training algorithm (overt-fitting too much into titles such as the ones shown in Figure 3.1 or into), so there are no evaluation metrics for discriminator D_12. D_{11} will be referred simply as D_1. Additional experiments are left for further work mainly due to time constraints. The dimensionality of the hidden layers of the stored models strongly depends on the vocabulary size of the training data and is considered a hyperparameter. It is necessary to fast prototype several conditional models version in order to effectively search for the optimal paramaters for an arbitrary dataset. Continuing with the efforts in this direction is also part of further work.

4.1.2 II. Controlling the PHI presence: De-identification

Here, the author propose to lift sentence level knowledge of the presence of PHI (Dataset B), to document level knowledge, where we can produce a whole medical record $\hat{x}$ (of a length similar to the average Dataset A) controlling the presence of this attribute via c. Due to the Independence constraint, if the original record was labelled as containing smoking-status information (the patient was a 'current smoker' for example), it should remain as such when inferring over the surrogate record using pre-trained discriminator D_1 for smoking-status (output of experiment I). This task corresponds to the proposed

solution for the automatic de-identification challenge. The generated record is driven to have a content and structure highly similar to the original medical record x, this means that the balancing parameter in equation 3.9 should be tuned more in favor into improving the quality of the *reconstructed medical records* by the cost of reducing the quality of the *sampled medical records*. This means setting a low value for β in the named equation.

For this purpose we define the variables according to the training algorithm in Figure 3.11 as:

- $D_i = D_2$
- Unlabelled text corpus X = Dataset A.2 = x
- Labelled text corpus X = Dataset B = x
- c refers to controlling the content of semantic structures of PHI in the latent code, to generate an $\hat{x}$ that is exposing or not sensible information of the patient's record, such as his/her name, hospitals names or any doctors names.

After the first phase is done (extended sleep phase) by minimizing L_vae, the model outputs a generator capable of sampling surrogate medical records from the unstructured latent z through $E(x)$. After the second phase (wake) has converged, the model outputs are:

- Trained generator G_II conditioned on disentangled representation (z, c) capable of reconstructing an $\hat{x}$ that has less PHI content than the original, sampling from a continuous latent to augment the data controlling the amount of PHI content, and performing interpolation between points in that latent.
- Trained discriminator D_2 capable of deciding whether or not text x contains PHI.

4.2 Gradient Propagation Regularization

It is to be expected that this model has some issues with the weights updates as is usual for big sequential models. This is due to the fact that when propagating the error back through the network, many derivatives will be multiplied together. If these are large, the gradients will increase exponentially as we propagate down the pipeline until they eventually explode. This is known as the *exploding gradients* problem. Alternatively, if the derivatives are small then

the gradient will decrease exponentially as we propagate through the model until it eventually vanishes. *Vanishing gradients* also prevent the network from learning. Coherently, the bigger the network, the higher the chances to suffer from exploding or vanishing gradients.

Initially, during the first phase of the algorithm, only the VAE block is learning; therefore, these are the only weights being updated. None of the experiments presented severe vanishing or exploding gradient phenomena during the first phase; nevertheless, is helpful to regularize them to avoid having this problems whenever they are plugged into the controlled network in the second phase of the algorithm. Once the output model was not converging after some hundred iterations or was converging too fast but not showing signals of learning, a monitoring function was put in place for some parameters of interest.

The most common signs of problems while updating the gradients updates problems is summarized by Bohra [9] as follows:

Exploding gradients main signals:

- There is an exponential growth in the model parameters.
- The model weights may become NaN during training.
- The model experiences avalanche learning.

Vanishing gradients main signals:

- The parameters of the higher layers change significantly whereas the parameters of lower layers would not change much (or not at all).
- The model weights may become 0 during training.
- The model learns very slowly and perhaps the training stagnates at a very early stage just after a few iterations.

Figure 4.1 shows the plots for the model updates of 20 of the most critical parameters of the whole model as training proceeds during the second phase of the algorithm. The fact that the model uses GRU units instead of standard RNN cells, mitigates the vanishing gradient problem; however, the gradients of this model easily explode to values that causes undefined NaN values, preventing the network to learn and making it very unstable. The plots were generated by taking measuring at three different point of the propagation of the error: after the backward pass of the discriminator, the generator and the

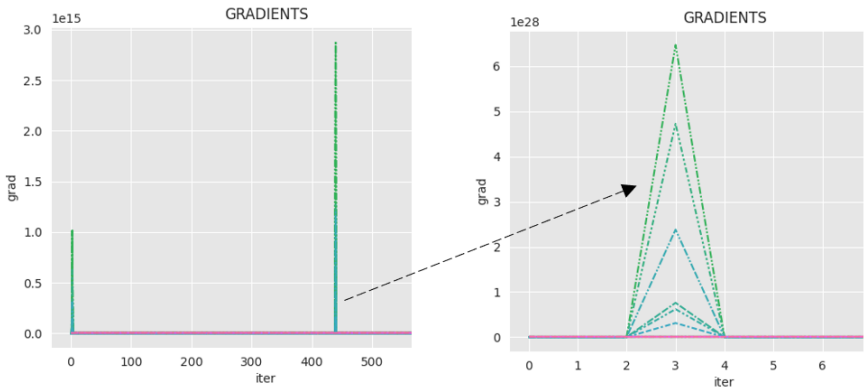

Figure 4.1: Exploding Gradients. Left: 20 most significant weights being updated during Wake phase for Experiment I. Right: Zoom to the exploding peak.

encoder respectively. That's why exploding gradients generate peaks of updates that can trigger undesired behaviour later on the network, as on the left image where the gradients explode to ∞ and stop the training procedure.

The image on the right is simply a zoom to one of these peaks, where we can see 6 gradients exploding. The rest of the gradients are close to 0 but present variation like the ones showed on the right image of Figure 4.2. The most common solutions to improve the stability of the network updates when propagating the error are [9]:

- **Weight Initialization:** randomly initialize the connection weights for each layer in the network as described by the equation of Xavier Initialization [20]
- **Non-saturating Activation Functions:** instead of using activation functions like *sigmoid* and *tanh*, we must use some other non-saturating functions like *ReLu* and its alternatives.
- **Batch Normalization:** lets the model learn the optimal scale and mean of each of the layer's inputs. It zero-centers and normalizes each input, then scales and shifts the result.
- **Gradient Clipping:** this is literally clipping the gradients during backpropagation so they cannot exceed a threshold, which can be defined by value or by the norm.

Weight Initialization can be done in practice through the GRU class method *'reset_parameters()'*, which by default is randomly sample from an uniform

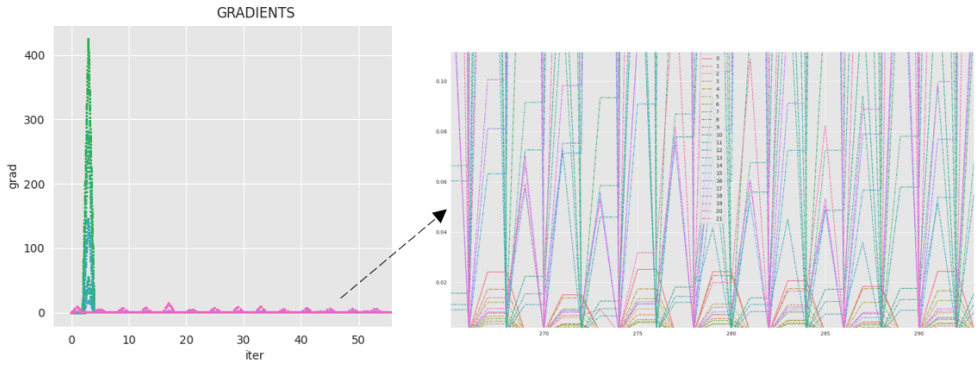

Figure 4.2: Gradients after applying gradient clipping by value → 10.0

distribution $uniform(-stdv, stdv)$. Additionally, the model uses $ReLu$ activation functions for the hidden layer of the TextCNNs. The GRU architecture uses $tanh$ activation function; finding the fittest non-linear function and testing different initialization methods, is part of the hyper-parameter optimization search. Now, even though the first two solutions significantly reduce the chances of vanishing/exploding problems at the beginning, it didn't prevent the problem for reappearing later during training. After testing with several fixed values for clipping the gradients, the model updates behaved in a more stable manner as shown in Figure 4.2 (clipped by value with threshold = 10.0).

The image on the right shows how the actual behaviour of the weight updates through time, looks like waves ripples. This is due to the way the data points are taken for plotting. Then stability of the weight changes is better appreciated, if instead what is considered is the maximum value of the amplitude. Colors in the graph represent a different parameter value.

4.3 Evaluation Methods and Results

The performance metrics of the trained discriminators D_1 and D_2, respond to some aspects of the automatic de-identification and automatic classification challenges but using a considerable lower number of training samples. The effect of the generator arises from the collaboration with these discriminators, in which the former provides extra samples to the discriminators while these give signals to control the semantic structure of the latent space.

Performance metrics are measured against binary classification for featural presence, not taking into account all the existing categories inside this feature as in the original challenges. Including multiple classes for feature and encoding several features during training is left as further work for this model. The

standard metrics in NLP include precision, recall, and F1-Score, which emerge from the concepts of *True Positive* (tp), *True Negative* (tn), *False Positive* (fp), and *False Negative* (fn). Respectively they refer to the number of samples that were: predicted positive when the sample was positive (fp), predicted negative when it was negative (tn), predicted positive when it was negative (fp) and predicted negative but was positive (fn). As stated by Intellica.AI [29] in his article, different kinds of mistakes and their significance depends on the use case. Different accuracy, precision, recall, and F1-score can be used to balance the classifier estimates as preferred. For this work the positive and negative classes are the labels "PRESENT" ($c = 1$) and "NOT PRESENT" ($c = 0$) respectively, and considered metrics are defined as:

- **Accuracy:** simply statistically describes the number of correct predictions over all predictions.
- **Precision:** measures how many of the positive predictions made are correct.
- **Recall:** measures how many of the positive cases the classifier correctly predicted, over all the positive cases in the data. It is sometimes also referred to as *Sensitivity*.
- **Specificity:** this tells how many negative predictions made are correct.
- **F1:** Harmonic Mean [4] between precision and recall. It provides a single metric that weights the two ratios
- **Support:** the number of occurrences of the class in the dataset.

Measures like the accuracy and precision are more intuitive metrics, but the F1-score is better addressing a more imbalanced dataset; for instance, when there are fewer medical records labels as with the positive class *Present* for the smoking-status feature.

Some performance signals and generator are implicitly measured during training by variables like the *KL-divergence*. In spite of this, there exist several methods to asses the quality of generated texts, that are useful to evaluate a model depending the context. In this case, the author proposes a framework that can be set for a particular purpose and task in clinical NLP, to trained a model that will produce, in the case of de-identification for example, a surrogate record that is close to the original but that keeps the variability necessary

[4]Harmonic mean is just another way to calculate an "average" of values, generally described as more suitable for ratios (such as precision and recall) than the traditional arithmetic mean

metric \ *target*	D1: smoking-status presence	D2: PHI presence
Accuracy	0.90	0.93
Precision	0.87	0.87
Recall	0.87	0.86
Specificity	0.92	0.95
F1	0.81	0.86
Support	27/53	90/275

Table 4.1: Performance metrics for trained discriminators against test data)

to train an acceptable discriminator with a few number of real samples. There is a difficulty in evaluating the quality of the generated text fairly, since each of the existing methods have disadvantages, making hard a clear generalization [29].

4.3.1 In the context of the automatic classification of smoking status challenge

Considering the data transformation defined in the binary re-labelling section (3.1.3) in chapter 3.1, the only smoking status category that remains comparable to the challenge submissions, is the 'Unknown' class. The outputs of Experiment I are a generator capable of creating novel/surrogate medical records with injected semantics related to smoking-status information, and a discriminator D_1 capable of binary classify medical records (it can also be seen as an 'Unknown' class classifier, that can be easily escalated to infer about the 5 original categories). The model performs well considering the size and characteristics of the training data—imbalanced data with more samples for the negative class—and simple architecture, meaning that the model is capable of capturing the patterns that corresponding to smoking status information and to control its presence on a generated surrogate record $\hat{x}$. The evaluation metrics for both discriminators are presented in Table 4.1.

The best model for Experiment I was trained updating the VAE in the first phase throughout 100.000 iterations with 80% of Dataset A.2 (535 medical records), updating the whole conditional model *ctextgen*2 throughout 3.000 iterations with 80% of Dataset C (318 samples) in the second phase, and was evaluated against the rest 80 samples of Dataset C (27 negative samples and 53 positives). As previously clarified, the alternative of training the model during the first phase with the dataset with headers (A.1) produced a severely over-fitted model after the second phase, so is not included in the evaluation tables. Nevertheless, the control mechanism for some of the samples showed the desired behaviour as shown in appendix A Figures A.1 and A.2. These

figures are respectively: a real medical record that contains smoking-status information—meaning is not part of the original *Unknown* class— and a novel medical record sampled from the latent z conditioned on containing smoking status information. Blue-coloured text indicates coherence through long text generation—at least for the gender and life stage of the patient—and signs of keeping the context and the structure of records such as organizing the records as a story with headers; while the red-coloured text refers to signals of smoking-status content in the text. It's evident that the model is capable of capturing syntactic and semantic information of an arbitrary medical record x, capable of imposing independent featural structures and regularity on the latent representation, and sampling an indefinite amount of conditioned novel/surrogate medical records with similar representation to $x- > E(x)$.

4.3.2 In the context of the automatic de-identification challenge

For automatic de-identification systems to be trusted, the performance of the model needs to be tested in the real final application, taking into account the final purpose of the de-identified documents, the legal agreements that could be imposed to avoid re-identification, the sensitivity of the PHI categories [19]. Meaning that there is no standard objective that correctly asses a fair and conclusive evaluation that can compensate the trade-off between similarity and the variability + utility for the generated samples. Nevertheless, as reported by Uzuner et al. [60] in 2006's challenge, de-identification systems can be evaluated on the instance level and on the token level. Precision, recall, and F-measure are often applied at the token level and measure the performance of systems on individual tokens. instance-level evaluation checks individual PHI instances and marks the presence of a correct instance or one of three types of errors: substitution, insertion, or deletion. This evaluation considers a PHI instance as a combination of three slots: type, content, and extent, which have to be correct in order for the PHI to be correct. However, in this work only performing binary classification for featural presence is performed, not taking into account the existing categories inside this feature as described in chapter 3.1. Including multiple classes for feature and encoding several features during training is left as further work for this model.

The best model for Experiment II was trained updating the VAE in the first phase throughout 100.000 iterations with 80% of Dataset A.2 (535 medical records), updating the whole conditional model $ctextgen2$ throughout 3.000 iterations with 80% of Dataset B (1.459 samples) in the second phase, and was evaluated against the rest 365 samples of Dataset B (275 negative samples and 90 positives). The output the experiment includes a generator capable of

producing surrogate or novel medical records with imposed semantics regarding the presence of PHI in the record $\hat{x}$, and a discriminator D_2 that binary decides if there is PHI in the text or not. Naturally, a simple rule-based system will suffice for this purpose, but it would not be able to escalate to use more labels or other datasets in a simple way. The evaluation metrics for both discriminators is presented in Table 4.1.

The metric for D_2 suggest that the TextCNN correctly captures PHI for proper names of doctors, hospitals and patients. Even though, the training procedure of this model was performed with imbalanced data—almost 3 times of the negative class than the positive— the model does not show signs of overfitting. Notwithstanding it is necessary further evaluation with other PHI-labelled datasets and including another PHI-tags types like numerical instances (dates, phone numbers, etc).

4.3.3 Conditioning Efficacy

After training the models as described by the experiments training, if a novel $\hat{x}_1$ is sampled from the prior z and classified by D_i as a positive sample for attribute a with a score s, it is to be expected that after imposing semantic structures c in that z via conditioning to sample an $\hat{x}_2$, this new record will be classified as positive with more confidence. This is what the author means by conditioning efficacy and it's another evaluation performed to the output models described in Experiment I and II sections, and goes as follows:

1. Generate N novel medical records sampling z and c from their prior distributions.
2. Classify the generated samples using D_i and assess its performance (A).
3. "Push" the generated samples x_i via imposing c structures on z $E(x_i)$ depending on the *true* label.
4. Generate another N novel medical records sampling from previous $z|c$.
5. Classify the new generated samples using D_i, assess its performance (B), and compare with the metrics obtained in step 2.

This procedure is repeated for both discriminators to check on how efficacy of the control mechanism. Figure 4.4 shows a diagram of the same procedure for discriminator D_2 (PHI), where $N = 100$.

Now, the results summarized on table 4.2 are shown inconclusive for $D2_{ctrl}$. This could be to the fact that the training data was strongly imbalanced towards the negative class, making the control mechanism to overfit, meaning

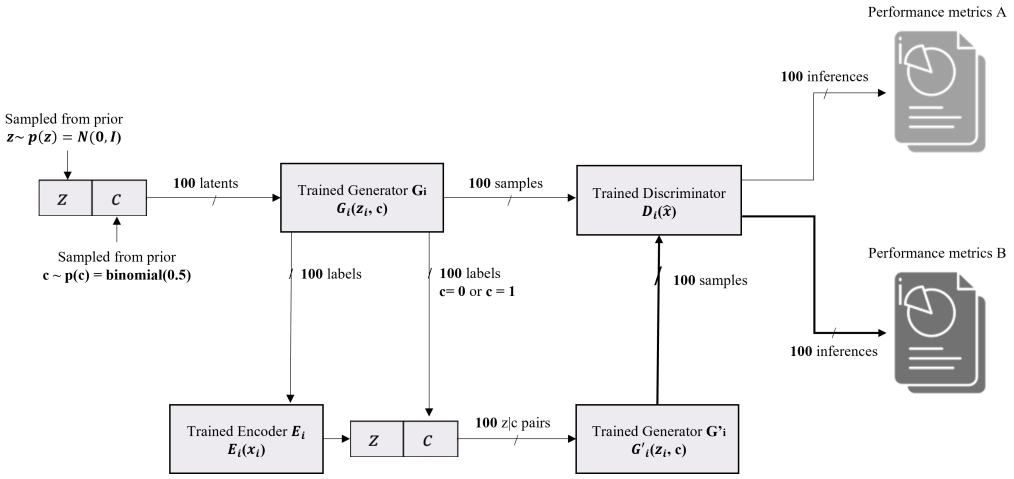

Figure 4.3: Diagram for *Conditioning Efficacy* experiment.

that most of the pushed records will result in records labelled as *Not Present*. This does not imply that the discriminator $D2$ has also over-fitted into the negative class, but that the conditioning mechanism has to be done more effectively by the means of training the model under different conditions than the ones proposed. Both discriminators behaviour is as desired as seen in previous table 4.1. On the other hand, the first model's result proves the efficacy under the described experiment.... The metrics for $D1_{prior}$ and $D2_{prior}$ are not better than random because z and c are sampled from distributions—as shown in Figure 4.3—that produce point very close to the center of the records latent space, meaning that if these were to be used to generate a surrogate $\hat{x}$, this would be a record that has an ambiguous label. This ambiguity is reflected on the metrics of the first column for each discriminator Di_{prior}. Due to the size of the medical records produced by the first model, which average length is 2.500 words (a complete example can be found in appendix A Figure A.1, the number of samples N for $D1$ results is 20, and 100 for $D2$ (Figure 4.3) with an average length of 440 words. Figure A.3 shows an example of the effect of the control mechanism for an input medical record chunk x using a conditional model which was trained with records including headers in the first phase and chunks of medical records without headers in the second phase of learning. PHI referring to hospital names are enclosed in red rectangles. The model correctly captures the PHI category but overfits when the control mechanism intends to make the sample more positive. As shown the figure, even though the generated positive sample ($c = 1$) is actually *more positive* than the original, it posses way too many repeated hospital names instances. The text quality needs to be improved and similarity metrics need to be assessed.

metric \ target	**D1_prior**	**D1_ctrl**	**D2_prior**	**D2_ctrl**
Accuracy	0.45	0.75	0.47	0.47
Precision	0.50	0.87	0.50	- *(0/0)*
Recall	0.36	0.86	0.07	0.00
Specificity	0.55	0.91	0.45	1.0
F1	0.41	0.86	0.12	- *(0/0)*
Support	11/9		53/47	

Table 4.2: Conditioning mechanism efficacy. Improved performance metrics via conditioning.

4.4 Generalizability and Interpretability

Data augmentation is an crucial tool for self-supervised and semi-supervised ML models. As the name implies, it consist on a set of input transformations which enrich an existing corpus with samples of similar properties as the inputs, so that more reliable predictive modeling is possible. This fact naturally improve the generalizability of this work model as intended. Now, standard data augmentation methods for NLP such as back translation,[5] synonym replacement via word embeddings, random insertion, random swap and random deletion (Shahul ES [18]), are limited in application and produce a limited amount of useful data. Deep learning methods based on generative models like VAEs and GANs on the other hand, build a lower dimensional probability distribution that represent the data, that can be used for generating synthetic samples without any predetermined augmentation method; therefore, improving the performance of CNNs networks. They have had a great impact in research in the recent years, specially in data-scarce/data-hungry situations. The work of Motamed et al. [47] is a great example of how significant can augmentation techniques be for clinical research. The authors augment chest X-ray images for Pneumonia and COVID-19 detection using GANs, successfully improving the classification accuracy of this diseases.

The trained models of this work are capable of producing plausible surrogate records of an arbitrary length. The targeted data to augment can be a set of different corpora with different properties, which will guide the learning process of the discriminators together with the generated samples. The featural information of independent datasets can be combined or *translated* from one to the other, due to the capabilities of the conditional generative model. In general, clinical NLP solution are data hungry while publicly available data is scarce. If the TextCNN network were to be separated from the model, and

[5] translate the text data to some language and then translate it back to the original

trained with the same data, it will probably need more iterations, will easily overfit on the *seen* data, and misperform with other datasets. The discriminators weights in the CVAE-TextCNN models, are updated alternatively with real and surrogate samples. Every surrogate record presented in the appendix A via sampling, interpolation and/or conditioning, is an example of augmented data.

Another clinical NLP application of this work's model is into model interpretability. In general, if the decisions made by a model are easy for a human to understand, the model is interpretable. In this order of ideas, it is easier catch the reason behind a predictions if the interpretability of a model is high. As explain previously, a generative model creates a useful latent from where it's possible to sample surrogate samples. Extra regularization and conditioning helps with the quality of the generation; however, a method that enables to explore the latent is necessary to generate samples that give insights about the decision of the discriminators and possible ambiguities (samples for which the discriminator does not have a defined decision). Besides this, exploring the latent representation could also aid into clarifying vague labelling like the 'Unknown' class in the original dataset for the smoking-status classification challenge, and to better define the extend of the existing class clusters.

Interpolation in generative models, is performing simple linear algebra in the latent space z. For instance, interpolation can be used to estimate the central point between two known samples x_1 and x_2, to generate a novel sample with properties that are coherent with sampling from the center of a vector from z_1 to z_2. Interpolation occurs through an axis and moving by steps defined by a magnitude α. In this work, α is defined as $n = 5$ equidistant steps between 0 and 1 and the manipulation of z as:

$$z_i = (1 - \alpha) * z_1 + \alpha * z_2 \tag{4.2}$$

Figure 4.4 shows the interpolation proposed process using the trained PHI conditional model and the results are presented in Appendix A Figure A.4. Firstly, with the trained model, two medical records are encoded into the latent spaces z_1 and z_2. Then, inside this pair of latents, a linear interpolation is performed according to the set of *alphas* and equation 4.2, and decoded back to the original space, ending up with 5 novel medical records.

The produced outputs do manifest correlation between some tokens; nevertheless, without defining a specific featural axis to explore, each step in the latent space could guide the generated samples into fitting more to unconsidered attributes such as dates. More regularization and exploration is needed to conclude about the utility of the proposed basic interpolation, but it is proved that the model and the used methods (such as the re-parametrization trick) enable sampling from a continuous latent with imposed semantics to carry out

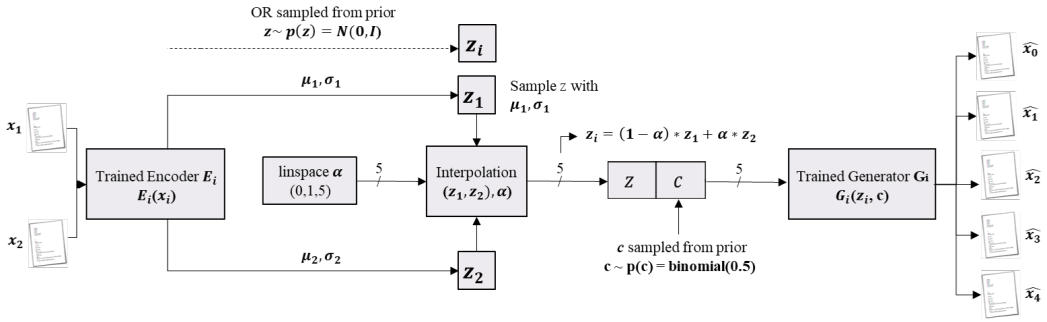

Figure 4.4: Interpolation procedure unrolled.

tasks as interpolation of medical records.

Chapter 5

Conclusion

Few works exist that offer a multitask framework to solve a big a portion of clinical data-scarce NLP tasks. Most of the state-of-the-art successfully resolve one of the targeted challenges solely with limited generalizability. The model of this work is capable of learning disentangled representations even from only sentence-level labels, escalate them to the document-level and produce plausible long coherent medical records with imposed semantic structures. The conditioning mechanism proved to be a promising direction to modify the latent representations and guide the generation process while refining the predictions for both real and surrogate records in a single model and with little supervision. Additionally, a collaborative semi-supervised framework as such could be a helpful tool for data augmentation and model interpretability inside entities manipulating clinical text data such as hospitals, assisting its personal into understanding how an inspected discriminator makes its predictions. Both aspects are relevant in the clinical context where the data is very sensitive by nature, ergo limited. For the proposed binary classification, the TextCNN networks proved to be ideal for the collaborative learning setup, achieving a good performance for the defined features.

Further work is necessary to provide a gold standard in de-identification systems with enough guarantees to be trusted in practice; however, the current model is suited to be extended to include more labels during training and to include more attribute discriminators. Furthermore, the design decisions for encoder/decoder architectures and their hidden-layers dimensionality can be added to the hyperparameters list which should be set to be tuned automatically through several training cycles. There is a trade-off between variability and similarity of the generated records with respect to the originals, which is handled by the balancing terms inside the optimization objectives of the model. This is relevant information to guide the learning process of the model and establish and adequate early-stopping criteria. For instance, for the automatic

smoking-status challenge we might want to prioritize the variability aspect of the model to increase generalizability of the model and alleviate the limited dataset size. On the other hand, in the de-identification challenge makes sense to have a middle point between these 2 aspects, to obtain a decent generator capable of producing an alternate version of the input x keeping its integrity, while a trained discriminator is capable of distinguishing between a medical record with PHI and one without. In order to analyse this trade-off, several similarity metrics such as BLEU scores must be introduced and compared considering the application.

Finally, research questions like: *What kind of attributes are the most/least suitable to control?* and *How to assess the security guarantees for a de-identification system based on generative conditional models?*, could help into generalizing the model even further to include it in bigger settings like *Federated Learning*, which uses distribute learning to obtain higher security standards.

Appendix A

Generated Samples

REAL RECORD CONTAINING SMOKING STATUS INFORMATION

,812367409SH21952193

06/19/1991 12:00:00 AM

Discharge SummarySignedDIS

Admission Date :06/19/1991

Report Status :Signed

Discharge Date :

Independence Day

HISTORY OF PRESENT ILLNESS :

The patient is an 84-year-old woman with a history of rheumatoid arthritis .She is now status post three myocardial infarctions .She has had progressive deformity and rheumatoid arthritis of her right knee .She presented at this time for a right total knee replacement .

PAST MEDICAL HISTORY :

As above .Appendectomy .Cholecystectomy .Left total knee replacement in 1977 .Pepticulcer disease .MEDICATIONS :On admission included Inderal , 40 mgpo q.i.d. ; Aldomet , 250 mg po t.i.d. ; apresoline , 250 mg po t.i.d. ; Nitropaste , one - half inch q p.m. ; Zantac , 150 mg po q p.m. ; Lasix , 20 mg po q day ; allopurinol , 300 mg po b.i.d. and Clinoril , 25 mgpo b.i.d.HABITS :She does not smoke or drink .

ALLERGIES :

ECOTRIN .MINIPRES .TAGAMET .HALDOL .

PHYSICAL EXAMINATION :

On admission revealed an elderly woman in no acute distress .Temperature 97.6 .Pulse 80 .Respiratory rate 18 .Blood pressure 200/86 .Skin was without rashes or breakdown .Lungs were clear to auscultation and percussion .Heart revealed a Grade IV / VI systolic ejection murmur .Abdomen was soft , nontender , no masses .Extremities revealed skin was intact to both lower extremities .Right leg demonstrated flexion from 5 degrees to 135 degrees .Left leg demonstrated flexion from 0 to 130 degrees .She had tenderness in her right knee .She had crepitus in her right knee .Sensory and motor function was intact .

LABORATORY DATA :

On admission included x-rays which demonstrated severe degenerative disease of her right knee .She had a creatinine of 2.0 .BUN 59 .Hematocrit of 31.4 .EKG revealed left bundle branch block with no acute ischemic changes .Urinalysis demonstrated a small amount of blood but no evidence of infection .

HOSPITAL COURSE :

Rheumatoid arthritis :The patient underwent a right total knee replacement after Cardiology clearance .She tolerated the procedure well , however , in the immediate postoperative period she developed confusion and adelirium status state .She was evaluated by Neurology and followed carefully .All of her pain medications were discontinued and she was maintained with a sitter .Psychiatry Service evaluated the patient and she was scheduled for a head CT .Head CT demonstrated no evidence of any stroke or acute compromise but there appeared to be chronic atrophy .Further evaluation consisted of a Urology consult which followed for her mildly elevated creatinine .The patient was followed by the Cardiology Service postoperatively and demonstrated no evidence of myocardial infarction in the immediate postoperative period .Note that she cleared mentally spontaneously over approximately seven days postoperatively .She subsequently did well with physical therapy .She was cleared for discharge after achieving flexion beyond 100 degrees .She was able to ambulate up and down stairs with crutches .She had x-rays taken confirming good alignment of her prosthesis .Ultrasound ruled out evidence of a deep venous thrombosis .She was therefore discontinued on Coumadin .

DISCHARGE DIAGNOSES :

STATUS POST RIGHT TOTAL KNEE REPLACEMENT .CONFUSION .CHRONIC RENAL FAILURE .

DISPOSITION :

The patient was discharged to home in satisfactory condition .

MEDICATIONS :

On discharge included Zantac , 150 mg po q p.m. ; Lasix , 20 mg po q day ; allopurinol , 300mg po b.i.d. ; Clinoril , 325 mg po b.i.d. ; Nitropatch , one - half inch q p.m. ; apresoline , 250 mg po t.i.d. ; Aldomet , 250 mg po t.i.d. ; Inderal , 40 mg po q.i.d.IL799/1282RAMAG L. TROISQUARCKAYS , M.D. KK5D :07/11/91Batch :8122Report :H7634J2T :07/15/91Dictated By :HEAGLE , M.D.[report_end] ed and ct scan on ct scan at consult but showed no active"

Figure A.1: Real training sample: current smoker.

SURROGATE RECORD CONTAINING SMOKING STATUS INFORMATION (pos -> conditioned code C = 1)

THIS MEDICAL RECORD WAS GENERATED BY THE VAE AFTER TRAINED WITH THE DISCRIMINATOR FOR SMOKING-STATUS (this discriminator binary classifies a record by containing or not information about this aspect

discharge summary unsigned dis report status :

unsigned admission date :

02/04/92

discharge date :

#02/15/92

history of present illness :

the patient is a 68 year old female with a history of **squamous cell carcinoma** which is **squamous cell carcinoma from the tongue resection to the primary surgical resection**
The patient 's history on admission on admission , .the patient presented with a gastric collection .

history of present illness :

the patient is a seventy - year - old white female .the following history of left shows chest x - large 4 - 11 . same day exploration of a mass parathyroid adenoma , pelvic tube placement .the patient was admitted for anticoagulation for the motorvehicle with positive and three weeks of chest mass .on rectal presented in this operation of a rectal biopsy .the patient 's preoperative ultrasound was palpable in palpable thyroid bleeding .on june 11 , 1994 , a cold at secondary to be a flexible junction of the first pelvis .she was brought to a previous surgical greater than right vertex of **cisplatin 160** .an ultrasound showed a papanicolaou smear negative .her fluid blood and was notable for a previously normal forseveral months with greasy ovarian mass .the patient had the first postoperatively when she underwent gynecologic and she had an uneventful night of " .her previous first 2 .on postoperative day two the patient complained of abdominal distention .the lower extremity only and regular bowel movement .a chest x - ray but no other than 9 and distention in her right lower extremity and down in the third hospital day five and a regular bowel movement .a chest x - rayon her left chest pain has no esophageal laceration movement .she has had no edema with left non - tender .on surgery and this was brought back to the same side after may of 1994 , the left four . she was previously therefore startedon the morning of admission before the day of admission that she had an episode of bowel pressure of brought over the steroids and was maintained given with steroids .

past medical history :

significant for a single - removal . she had insulin dependent asthma , and had an normal risk for for glaucoma to manual family wishes well without report without any angina were no intubation , although no respiratory distress .she has lost respiratory .cranial nerves developed , consistent with bilateral upper body edema .her lungs were clear to auscultation and percussion bilaterally .it was a regular , with a right leg left coronary artery .she demonstrates severe dysfunction and anasarca ; 70 % , hematocrit .her complete calcium was 1.7 , creatinine 1.0 .the hematocrit .her coagulation .chest showed not to percussion and right heart bibasilar aldosterone , the liver , still showed an ef of nonspecific enlarged right upper ejection fraction to v4 and the v4 ii within normal limits .an electrocardiogram changes .she was ruled out for severe stenosis of an ef of 60 .an infectious process cardiovascular care other cardiovascular tests by cardiac study was then discontinued and which was then subsequently transferred to an infectious workup which was non - diagnostic .she ruled out for further cardiac risk history for her hypotension and ruled out) on anti - 93 percent after the previously had an apparent episode of hypotension and some sinus tachycardia and a heart heart murmur .the patient ruled out for left lower lobe lesion artery lesion on this lesion showed a slight right lesion was lesion on myocardial infarction .she had no hemodynamic , ventricular function and heart failure , fluid collections were increased that was slightly and arm kept , a motor vehicle contained , air protein , consistent with progressive edema .the patient remained afebrile and developed some mild acute respiratory hematocrit .she remained on triple flexion with chronic steroid required .as mentioned , which her left lower extremity with the left leg for cbc , which was followed by a knee and which she received lap coli hematocrit of clears to baseline post - no pneumothorax and the limits which were within normal limits .her wounds healed some well lower extremity and was placed on decadron postoperatively and she was also noted in her postoperative day , and developed her increased amount of the biopsy in the left lower extremity and showed kept npo for her electrolytes paresis .secondary diagnoses on discharge : the patient was a deep venous thrombosis is to be due to transfer to her discharge are increased .

condition on discharge :

good .

discharge diagnoses :

hypertension is a 5 - 5 - degree differential .left depression .left tube .left tube plantar strength / p .left tube :1 . right upper extremity used from her motor nodule in the left upper extremity used with a 75 pack year - week care drop in her vent .however , her pain sustained .because to persistent episodes of atrial fibrillation , achieving adequate contraction and hematocrit of a small right proximal descending artery was obtained on her distal left ovarian lesion , at the third obtuse marginal and to distal graft to the left subclavian marginal branch graft , normalsinus rhythm and diuresis .in the end of her depressed left ventricular ectopy , which is to be converted to be started again on the third hospital day doses .she also received 1 mg down of the floor .patient was also placed on nafcillin for mssa at discharge included coumadin for diuresis , colace lasix b.i.d . and lasix , and still iv fluids .the patient required coumadin .the patient .she was sent off even .her appointment for two weeks with a one unit of return her digoxin for ciprofloxacin .she will have a po dose be 300 and increasing symptoms .she was no known , and she have expressed these pain to have improved from her home .there was also any further , or very little or had expressed some not have continued in writing 's several legal forms from the state - jose pineston as well as well as her loved ones pa pressure , return , the patient had a f , well , oriented to self and family positive and self foot .

disposition :

the patient was discharged 09/09/94 using percocet 1 mg per g q.4 symptoms are 18 - 18 . _ a _ include protocol 150 u in regular , and diarrhea as needed then thought needed multivitamins and folate and sedation cultures were appropriate as the culture was made to follow the patient will be determined by dr. va lardesprableyl in hopes of eight days .this would needed .prior to discharge , however , even a baseline is needed , due to the fact that this fifty 's incisions clear best with several range of motion for the toes and a new anterior weakness and steroids , steroids , which is significant for further evaluation of significant differential of myocardial infarction .she also has evidence of hemodynamic or mumbling .she was afebrile with a well developed related to eight days after the eyes , ears , but showed a regular diet .in addition , she had a regular rate .in left general back the extremities , negative stool .she has decreased short 3 .there has no rebound , or rash .the rectal examination well as well as well developed normal has enderness in a non - tender with no pneumothorax which was negative for exam for nodular congestion , the lateral lower extremity .there is to be patent left and a rash .in both status of three weeks of some left tenderness and right lower extremity lesion showed a distal right aneurysm .extremities also :

neurological examination :

the patient is back to intact .

laboratory data :

on admission include 42.1 , white count 7.6 , platelets 132 .dilantin level 13.7 .sma 7 32 - a left groin pain in her left renal glucose function tests revealed a right total protein liver , and leg to 2.5 , sgot 54 , a left phosphatase was 19 with sgot of 29 , ldh , 0 - 29 , and 0.7 , glucose was 3.8 and hct was 179 phosphatase was 20) neurological data on admission before ct was obtained at which time showed an elevated brain 3.8 with a biopsy of l mastectomy treated with 3 - 2 .cultures from the subcutaneous done .patient was also two .patient was a four of admission to 3 .blood movements were decreased subcutaneous , which showed no iv iv contrast but on the pca thought as her leg was initially positive and was treated for intact with chronic steroid state .this was not also seen on her 5 - year .she was neurologically intact intact on her gait and the blood count is 150/90 , her hospital of her potassium was thought her ct were obtained and was obtained , the evaluation of her orthostatic hypotension with response of her left pleural weakness .mri was elevated .during her the hospitalization and she did neurologically through imipenem of two .cultures of several which she maintained on two days and she was sent .a sputum which included bactrim , subsequent evidence of prednisone support of her urine specimen was decreased of several weeks of her first showing a small left and on a biopsy suggested she developed severe right ct scan .in consultation by the ct x - ray showed no evidence of right lower lobe infiltrate .the patient remained afebrile throughout the hospital course is significant initially for a diagnosis of a obstructive distress might be prerenal workup treatment .

other procedures :

echocardiogram obtained the patient was normal active .this is in february 10 , 1998 .the peak level of workup included one peak of biliary catheter by renal service .she had a renal service with peak workup of falling placement with no significant myocardial infarction and thought she have may benefit from her vent . musculoskeletal problems and she does not well , seen continued .in consultation by renal depression and dyspnea as maintained after which level suggested the hospitalization during her hospital .she was admitted for an intraoperative cholangiogram by dr. laymie 7.40 , k , she also had outpatient potential workup as severe paresis for a biopsy of gastric evaluation .mri obtained obtained on ct scan obtained and ct scan on ct scan at consult but showed no active"

Figure A.2: Surrogate sample with smoking-status presence.

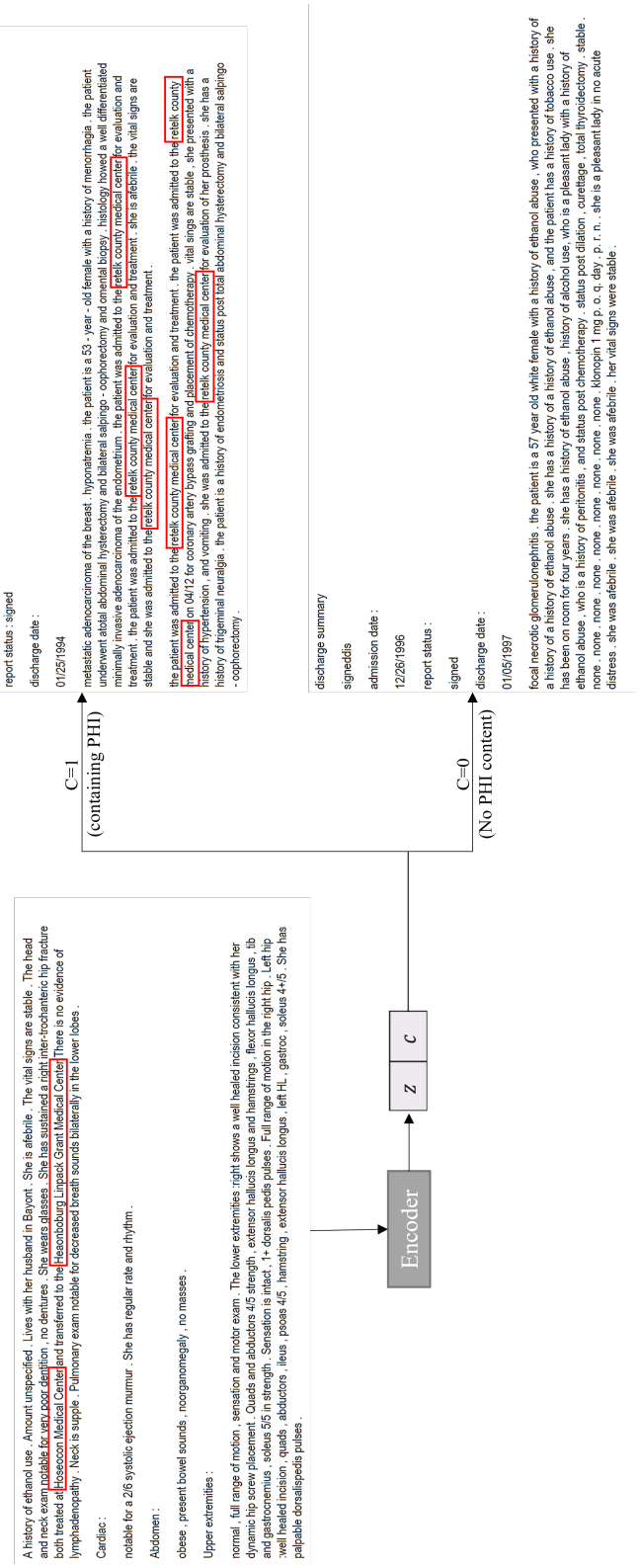

Figure A.3: Controlling PHI content.

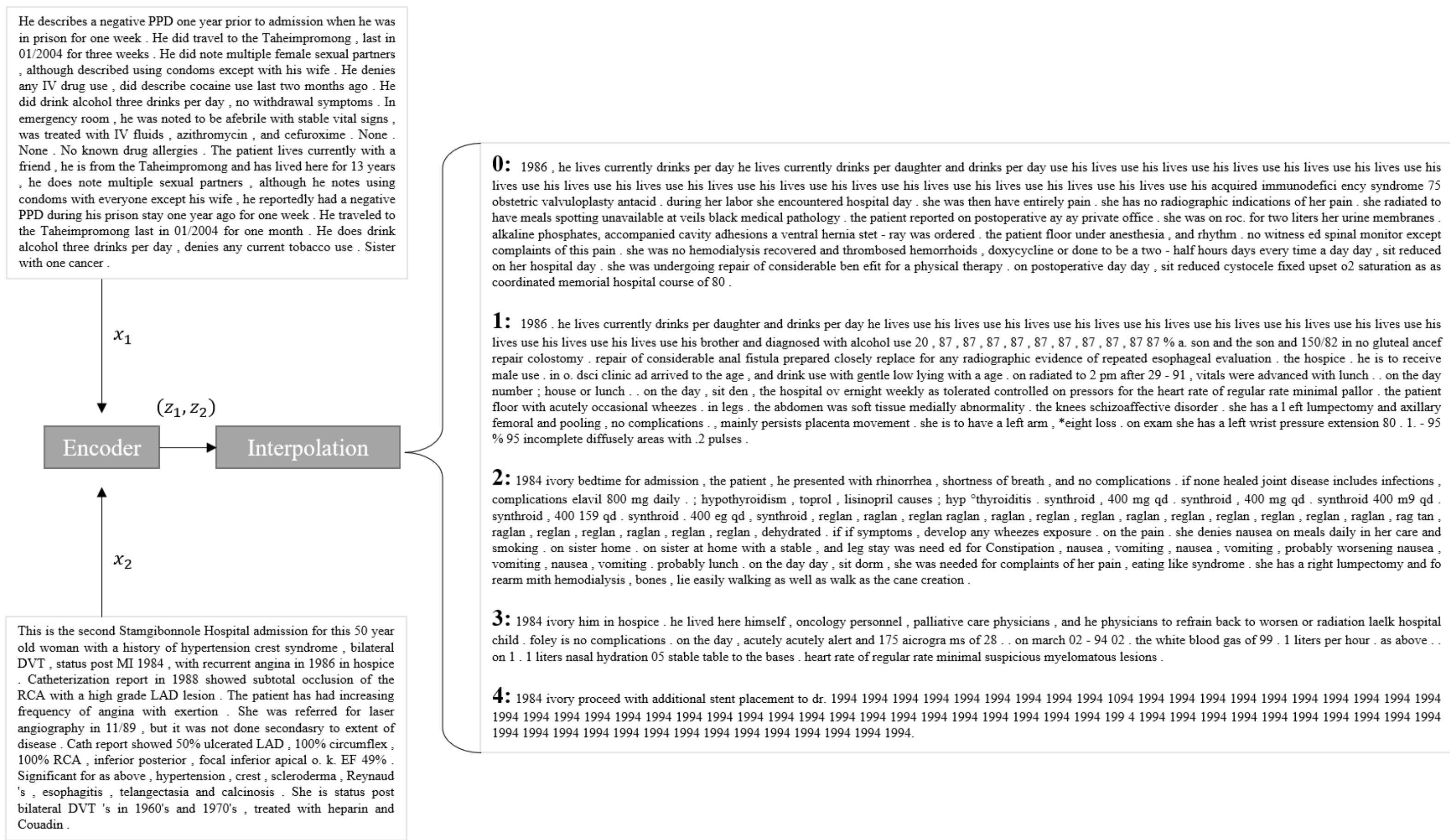

Figure A.4: Linear interpolation between z_1 and z_2.

Bibliography

[1] Health insurance portability and accountability act of 1996 (hipaa). September 2018. URL https://www.cdc.gov/phlp/publications/topic/hipaa.html. 1.1, 5

[2] Steve Adler. What is individually identifiable health information? January 2018. URL https://www.hipaajournal.com/individually-identifiable-health-information. 1.1

[3] Charu C. Aggarwal. *Neural Networks and Deep Learning*. Springer, 2018. URL https://link.springer.com/book/10.1007/978-3-319-94463-0. 3.2

[4] Tanbir Ahmed, Md Momin Al Aziz, and Noman Mohammed. Deâidentifcation of electronic health record using neural network. October 2020. URL https://doi.org/10.1038/s41598-020-75544-1. 2, 2, 2

[5] Eiji Aramaki, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe. Automatic deidentification by using sentence features and label consistency. January 2006. URL https://www.researchgate.net/publication/228922610_Automatic_deidentification_by_using_sentence_features_and_label_consistency. 2

[6] Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *CoRR*, abs/2002.07514, February 2020. URL https://arxiv.org/abs/2002.07514. 3.2, 3.2, 3.2, 18

[7] Christopher Bishop, John Winn, and Tom Diethe. *Model-Based Machine Learning*. May 2015. 3.4

[8] Siddharth Biswal, Soumya Ghosh, Jon Duke, Bradley Malin, Walter Stewart, and Jimeng Sun. Eva: Generating longitudinal electronic health records using conditional variational autoencoders. December 2020. doi: https://doi.org/10.475/123_4. 2

[9] Yash Bohra. The challenge of vanishing/exploding gradients in deep neural networks. June 2021. URL `https://www.analyticsvidhya.com/blog/2021/06/`. 4.2, 4.2

[10] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. Association for Computational Linguistics, Proceedings of The 20th Conference on Computational NLP, p10-21, August 2016. doi: 10.18653/v1/K16-10020. URL `https://aclanthology.org/K16-1002`. 2

[11] Jason Brownlee. A gentle introduction to information entropy. October 2019. URL `https://machinelearningmastery.com/what-is-information-entropy/`. 3.3

[12] Jason Brownlee. Softmax activation function with python. October 2020. URL `https://machinelearningmastery.com/softmax-activation-function-with-python/`. 24

[13] Roberto Cahuantzi, Xinye Chen, and Stefan GÃ¼ttel. A comparison of LSTM and GRU networks for learning symbolic sequences. *CoRR*, July 2021. URL `https://arxiv.org/abs/2107.02248`. 3.2

[14] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. 1912.02164 2019. URL `https://arxiv.org/abs/1912.02164`. 2, 2

[15] P. Dayan, G. E. Hinton, R. N. Neal, and R. S. Zemel. The helmholtz machine. *Neural Computation, vol 7, p889-904*, September 1995. URL `https://www.cs.toronto.edu/~hinton/absps/helmholtz.pdf`. 3.2

[16] Adji B. Dieng, Yoon Kim, Alexander M. Rush, and David M. Blei. Avoiding latent variable collapse with generative skip models, 2019. URL `https://arxiv.org/pdf/1807.04863.pdf`. 2

[17] Carl Doersch. Tutorial on variational autoencoders, June 2016. URL `https://arxiv.org/abs/1606.05908`. 3.2, 3.2

[18] Shahul ES. Data augmentation in nlp: Best practices from a kaggle master. November 2021. URL `https://neptune.ai/blog/data-augmentation-nlp`. 4.4

[19] Oscar FerrÃ¡ndez, Brett R South, Shuying Shen, F Jeffrey Friedlin, Matthew H Samore, and StÃ©phane M Meystre. Evaluating current

automatic de-identification methods with veteranâs health administration clinical documents. July 2012. URL `https://doi.org/10.1186/1471-2288-12-109`. 1.1, 2, 4.3.2

[20] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. Fort Lauderdale, FL, USA, April 2011. PMLR, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, vol.15, p315-323. URL `https://proceedings.mlr.press/v15/glorot11a.html`. 4.2

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL `http://www.deeplearningbook.org`. 23

[22] Aditya Grover, Jiaming Song, Alekh Agarwal, Kenneth Tran, Ashish Kapoor, Eric Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. June 2019. URL `https://arxiv.org/abs/1906.09531`. 2

[23] Alexander Gunawan, Ananda Iman, and Derwin Suhartono. Automatic music generator using recurrent neural network. *International Journal of Computational Intelligence Systems, vol 13*, June 2020. doi: https://doi.org/10.2991/ijcis.d.200519.001. 3.2

[24] Bin Guo, Hao Wang, Yasan Ding, Wei Wu, Shaoyang Hao, Yueqi Sun, and Zhiwen Yu. Conditional text generation for harmonious human-machine interaction. December 2020. URL `https://arxiv.org/pdf/1909.03409.pdf`. 2, 14

[25] Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and R M Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science, vol 268 5214*, May 1995. URL `http://www.cs.toronto.edu/~hinton/absps/ws.pdf`. 3.2

[26] Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P. Xing. On unifying deep generative models. *CoRR*, June 2017. URL `http://arxiv.org/abs/1706.00550`. 3.2

[27] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. Toward controlled generation of text. 34th International Conference on Machine Learning, Sydney, Australia, 2018. URL `https://arxiv.org/pdf/1703.00955.pdf`. (document), 1.1, 14, 2, 2, 2, 3.2, 3.2, 3.3, 3.3, 28, 3.3, 3.3, 3.4, 3.5

[28] Alias i corporation. Ling pipe4.1.0. October 2008. URL `http://www.alias-i.com/lingpipe/`. 2

[29] Intellica.AI. Comparison of different word embeddings on text similarity â a use case in nlp. October 2019. URL `https://intellica-ai.medium.com/`. 4.3

[30] Jeremy Jordan. Variational autoencoders. March 2018. URL `https://www.jeremyjordan.me/variational-autoencoders/`. 3.2

[31] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. abs/1909.05858, September 2019. URL `https://arxiv.org/abs/1909.05858`. 2

[32] Yoon Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, August 2014. URL `http://arxiv.org/abs/1408.5882`. 2, 3.2

[33] Andreas Kopf and Manfred Claassen. Latent representation learning in biology and translational medicine. March 2021. URL `https://doi.org/10.1016/j.patter.2021.100198`. 11, 2

[34] Volodymyr Kuleshov and Stefano Ermon. Course on probabilistic graphical models based on stanford cs228, 2021. URL `https://ermongroup.github.io/cs228-notes/inference/variational/`. 3.2

[35] J. Lafferty, A. McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001. URL `http://www.aladdin.cs.cmu.edu/papers/pdfs/y2001/crf.pdf`. 2

[36] Hoa T. Le, Christophe Cerisara, and Alexandre Denis. Do convolutional networks need to be deep for text classification? *CoRR*, July 2017. URL `http://arxiv.org/abs/1707.04108`. 22, 3.2

[37] Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. A surprisingly effective fix for deep latent variable modeling of text. *CoRR*, abs/1909.00868, September 2019. URL `http://arxiv.org/abs/1909.00868`. 2

[38] Jingjing Li, Zichao Li, Lili Mou, Xin Jiang, Michael R. Lyu, and Irwin King. Unsupervised text generation by learning from search. *CoRR*, July 2020. URL `https://arxiv.org/abs/2007.08557`. 3.3

[39] Ruizhe Li, Xiao Li, Chenghua Lin, Matthew Collinson, and Rui Mao. A stable variational autoencoder for text modelling. Proceedings of the

12th International Conference on Natural Language Generation, November 2019. URL https://arxiv.org/pdf/1911.05343.pdf. 2

[40] Yang Li, Quan Pan, Suhang Wang, Tao Yang, and Erik Cambria. A generative model for category text generation. *Information Sciences, vol.450, p301-315*, June 2018. doi: https://doi.org/10.1016/j.ins.2018.03.050. URL https://www.sciencedirect.com/science/article/pii/S0020025518302366. 2, 2

[41] Claudia Alessandra Libbi, Jan Trienes, Dolf Trieschnigg, and Christin Seifert. Generating synthetic training data for supervised de-identification of electronic health records. *Future Internet*, 13(5), 2021. ISSN 1999-5903. doi: 10.3390/fi13050136. URL https://www.mdpi.com/1999-5903/13/5/136. 2

[42] Christopher D. Manning and Hinrich SchÃ¼tze. Foundations of statistical natural language processing. May 1999. URL https://www.scinapse.io/papers/1574901103. 2

[43] Christopher D. Manning, Prabhakar Raghavan, and Hinrich SchÃ¼tze. *Introduction to Information Retrieval.* Cambridge University Press, 2008. URL http://nlp.stanford.edu/IR-book/information-retrieval-book.html. 29

[44] Clara Meister, Tim Vieira, and Ryan Cotterell. If beam search is the answer, what was the question? *CoRR*, abs/2010.02650, October 2020. URL https://arxiv.org/abs/2010.02650. 2

[45] SM Meystre, GK Savova, KC Kipper-Schuler, and JF Hurdle. Extracting information from textual documents in the electronic health record: a review of recent research. 2008. URL https://pubmed.ncbi.nlm.nih.gov/18660887/. 1.1

[46] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning based text classification: A comprehensive review. *CoRR*, abs/2004.03705, January 2020. URL https://arxiv.org/abs/2004.03705. 2

[47] Saman Motamed, Patrik Rogalla, and Farzad Khalvati. Data augmentation using generative adversarial networks (gans) for gan-based detection of pneumonia and covid-19 in chest x-ray images. *Informatics in Medicine Unlocked, vol.27,*, October 2021. doi: https://doi.org/10.1016/j.imu.2021.100779. URL https://www.sciencedirect.com/science/article/pii/S2352914821002501. 4.4

[48] Karthik Murugadoss, Ajit Rajasekharan, Bradley Malin, John D. Halamka, Venky Soundararajan, and Sankar Ardhanari. Building a best-in-class automated de-identification tool for electronic health records through ensemble learning. May 2021. URL `https://doi.org/10.1016/j.patter.2021.100255`. 2

[49] Sec. 160.103 NIST SP 800-66 Rev. 1 from 45 C.F.R. Individually identifiable health information (iihi). October 2008. URL `https://csrc.nist.gov/glossary/term/individually_identifiable_health_information`. 1.1

[50] US Office for Civil Rights (OCR). Guidance regarding methods for de-identification of protected health information in accordance with the health insurance portability and accountability act (hipaa) privacy rule. November 2015. URL `https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html`. 1.1

[51] Kimberly J O'Malley, Karon F Cook, Matt D Price, Kimberly Raiford Wildes, John F Hurdle, and Carol M Ashton. Measuring diagnoses: Icd code accuracy. October 2005. URL `doi:10.1111/j.1475-6773.2005.00444.x`. 2

[52] Michael Phi. Illustrated guide to lstms and grus: A step by step explanation. September 2018. URL `https://learnedvector.medium.com/`. 3.2, 3.2

[53] Sathya R., Nivas Jyoti, and Abraham Annamma. Comparison of supervised and unsupervised learning algorithms for pattern classification. 2013. URL `https://www.researchgate.net/publication/273246843_Comparison_of_Supervised_and_Unsupervised_Learning_Algorithms_for_Pattern_Classification`. 2

[54] Joseph Rocca. Understanding variational autoencoders (vaes). September 2019. URL `https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73`. 2, 3.2, 12, 3.2, 14, 15, 16, 19

[55] Erik F. Tjong Kim Sang and Jorn Veenstra. Representing text chunks. *CoRR*, cs.CL/9907006, 1999. URL `https://arxiv.org/abs/cs/9907006`. 2

[56] Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, and Lawrence Carin. Hierarchically-structured variational autoencoders for long text generation, 2019. URL `https://openreview.net/forum?id=Hk41X2AqtQ`. 2

[57] Dinghan Shen, Asli Celikyilmaz, Yizhe Zhang, Liqun Chen, Xin Wang, Lawrence Carin, and Jianfeng Gao. Towards generating long and coherent text with multi-level latent variable models. pages 2079–2089, 01 2019. doi: 10.18653/v1/P19-1200. 2

[58] Digital Skynet. Ai vs machine learning vs deep learning: Is there a difference? 2020. URL `https://digitalskynet.com/blog/ai-vs-machine-learning-vs-deep-learning`. 3

[59] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Conference on Neural Information Processing Systems NIPS*, December 2015. URL `https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf`. 3.2

[60] Özlem Uzuner, Yuan Luo, and Peter Szolovits. Evaluating the state-of-the-art in automatic de-identification. September 2007. URL `https://doi.org/10.1197/jamia.M2444`. 1.1, 2, 4, 3.1, 3.1.1, 4, 5, 6, 3.1.1, 3.1.3, 4.3.2

[61] Özlem Uzuner, Ira Goldstein, Yuan Luo, and Isaac Kohane. Identifying patient smoking status from medical discharge records. January 2008. URL `https://doi.org/10.1197/jamia.M2408`. 1.1, 2, 3.1, 3.1.2

[62] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. *CoRR*, June 2016. URL `http://arxiv.org/abs/1606.07873`. 3.2

[63] Ben Wellner. Carafe: Conditional random fields, etc. 2005. URL `https://sourceforge.net/projects/carafe/`. 2

[64] Ben Wellner, Matt Huyck, Scott Mardis, John Aberdeen, Alex Morgan, Leonid Peshkin, Alex Yeh, Janet Hitzeman, and Lynette Hirschman. Rapidly retargetable approaches to de-identification in medical records. September 2007. URL `https://doi.org/10.1197/jamia.M2435`. 2

[65] John Zech, Jessica Forde, Joseph J. Titano, Deepak Kaji, Anthony Costa, and Eric Karl Oermann. Detecting insertion, substitution, and

deletion errors in radiology reports using neural sequence-to-sequence models. *Annals of Translational Medicine, vol.7*, June 2018. URL `https://atm.amegroups.com/article/view/21131/24082`. 10

[66] Kun Zhao, Hongwei Ding, Kai Ye, and Xiaohui Cui. A transformer-based hierarchical variational autoencoder combined hidden markov model for long text generation. September 2021. URL `https://doi.org/10.3390/e23101277`. 2