UPPSALA
UNIVERSITET

UPTEC STS 200 26
Examensarbete 30 hp
Juli 2020

Building a Medical Recommendation System

A case study on digitalizing evidence-based radiology

Fabian Persson

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Building a Medical Recommendation System

Fabian Persson

In this thesis, we show how a text-based Recommendation Systems can greatly benefit from neural statistical language models, more particularly BERT. We evaluate the framework on a digital and collaborative platform for radiologists, by automatically suggesting scientific papers from the medical database PubMed, to provide evidence in diagnostic radiology. The models use contextualized vectors to represent text, accounting for writing style, misspelling and jargon. By using pre-computed representations of text passages, we are able to use compute-heavy statistical language models in production environments, where supercomputers are not available during inference.

The results suggests pre-computed embeddings are very effective when the texts came from the same domain, and less effective (but still useful) in capturing the interaction between clinical and scientific text. Nonetheless, the suggested solutions hold promises in this and other areas in medicine. Possibly, the results are transferable to other domains, such as processing of legal documents and patent search.

Handledare: Pär Kragsterman
Ämnesgranskare: Niklas Wahlström
Examinator: Elísabet Andrésdóttir
ISSN: 1650-8319, UPTEC STS 200 26
Tryckt av: UPPSALA

Populärveteskaplig sammanfattning

I denna uppsats genomförs en design och en implementation utav ett rekommendationsystem för praktiserande radiologer (medicinsk bildtagning). Utifrån autentiska patientfall från en digital och kollaborativ plattform (www.cmrad.com), skulle vetenskapliga artiklar föreslås, som förväntades vara behjälpliga för att ställa korrekt diagnos i det aktuella fallet. I allmänhet bör kliniskt och medicinskt beslutsfattande vara grundat på etablerad vetenskaplig evidens, men på grund av tidspress och ovana känner sig långtifrån alla läkare bekväma med att navigera stora medicinska databaser. Det riskerar att förhindra att etablerad forskning når ut till praktiserade läkare inom sjukvården.

För att bygga systemet användes olika metoder från fälten språkteknologi och maskininlärning. Löst definierat innefattar området språkteknologi datoriserade metoder för att processera text, och maskinlärning träning av modeller som utifrån inlärda mönster på historisk data ska kunna göra nya förutsägelser. Våra modeller tränades på data inhämtat från Open-I och PubMed, som tillhandahålls av den amerikanska federala organisationen *National Library of Medicine*. Huvudsakligen byggde modellerna på BERT, vilket är ett ramverk lanserat av Google för att träna språkliga AI-modeller med. BERT kan förstå kontext, nyans och synonymer - men också kan hantera felstavningar, genom att bryta ner ord till flera delkomponenter.

Målsättningen var att våra modeller skulle fungera i enklare miljö utan tillgång till superdatorer, varför vissa effektiva men väldigt beräkningstunga tillvägagångssätt i BERT valdes bort eller förenklades. Vi lyckades genom såkallade förberäknade representationer väsentligt förkorta tiden för modellen att göra nya förutsägelser med hjälp av BERT, från flera dygn till några sekunder på en vanlig dator.

Resultaten visade att om det språkliga domänet skiljde sig åt – i vårt fall klinisk text (produceras av läkare) och akademisk, peer-reviewad text – blev precisionen lidande vid användning av förberäknade representationer. Däremot lyckades vi visade att vi med väldigt goda resultat att vi kunde använda förberäknade representationer för att hitta liknande artiklar, till en redan föredragen artikel. En nyhet är att detta funkar lika bra även för längre textsegment. Vi tror att detta har flera applikationer inom en mängd andra domän, som nyhetsbevakning, juridik och patentsök.

Contents

1 Introduction

Radiology, also known as medical imaging, has assumed an important part of clinical medicine in the last decades. In between 20-30% of all patient cases, it is estimated patient symptoms alone are not enough to correctly diagnose the patient, requiring additional procedures such as medical imaging or lab tests. At the forefront of medical digitialization, Collective Minds Radiology operates a digital platform for practicing radiologists. The platform unites radiologists from different hospitals and countries to collaboratively establish the correct diagnosis or the right intervention, or share interesting cases for educational purposes. Analyzing medical images is often time-consuming and highly complex, especially so when radiologists encounter something they haven't been exposed to previously.

Ideally, medical decision making should follow well-designed research and clinical expertise [1, 2]. While there is an abundance of research that can possibly facilitate evidence-based practice in medicine, physicians and medical researchers often find the large volume of scientific research difficult and time-consuming to navigate [3]. Potentially, this may hinder academic research from entering clinical care.

In order to narrow the gap between research and clinical care, this thesis will implement a recommendation system of scientific papers in radiology. [1] The overall goal of the system is to accept written notes from patient cases and output relevant scientific papers from a medical database. We envision two core functionalities of the recommendation system:

1. The user may present the system with case postings from Collective Mind's digital platform, and in return get the presumed most relevant scientific papers back. We call this *case-to-articles*.

2. The user may also present the system with a scientific paper, and as a response get the most similar scientific papers back. Here, we reason that if a scientific paper x_1 is relevant to a case, and x_2 is similar x_1, we assume x_2 is relevant as well. We call this *articles-to-articles*.

[1] It should be noted however that the technologies underpinning this system will not be applicable to radiology only, but also applicable in other areas of medicine (or even completely different domains).

Formally, we are interested in learning a scoring function for each of the tasks defined above, so that we can a) in *article-to-article*, take a pair of scientific papers, or b) in *case-to-article* take a pair of a case and scientific paper, and for both scenarios assign a score on their estimated similarity or relevance respectively.

We will try to answer these question using late advancements from the intersection between the fields of Natural Language processing and Machine Learning. In Machine Learning, patterns are learned from observed data, as opposed to manually coding down rules. Natural Language Processing (NLP) explores interaction between human language and computers. Historically, NLP has consisted of mostly hand-crafted rules and algorithms. However, the rapid advancement of deep methods in machine learning has driven NLP to become increasingly intertwined with Machine Learning, replacing human efforts with automated methods for learning effective representations of human text [4].

There are very few modern examples of building a recommendation system like ours, so we draw inspiration from Citation Recommendation System, where the goal of the system is recommended relevant citations for a text passage written by the user [5].

2 Theory

The theory section is divided into a brief background on machine learning and the various method underpinning our system, followed by Natural Language Processing. This is followed by an introduction of graph algorithms, which was used as a proxy for articles-to-articles similarity in lack of labesl. Finally, a short overview of metrics used to evaluate our system are presented.

2.1 Machine Learning

Machine learning is a scientific area studying the application of computerized methods to automatically learn patterns from observed data. The learnt patterns can be used to make future predictions on new data, or uncover new relations previously unknown to the human. Typically, machine learning is divided into supervised and unsupervised problems. Unsupervised problem are often not as well-defined, as they deal with knowl-

edge discovery, rather than learning from explicit labels.

Supervised problems on the other hand usually involve training the algorithm against some kind of defined objective - penalizing the model when it makes bad predictions while rewarding it for good predictions. Formally, we may want to learn the mapping from an input to an output

$$F : V \rightarrow W. \tag{1}$$

The goal is to generalize this function approximation to unseen data. Otherwise, we have just learnt to memorize our historical data. A common way to evaluate the generalisability is to divide the data into a training, validation and test set. In this approach, we leave some observed data out during different stages of the training, to test our predictions on. [6]. In the following subsections, we will present two supervised machine learning models.

2.1.1 Neural Networks

Deep Neural network (DNN) is a special type of a machine learning method, mostly used for classification problems. Deep neural networks are often formally referred to as multi-layer feed-forward neural network. DNN's consist of an input layer, a set of hidden layers and an output layer. Every layer consists of a set of neurons, as illustrated by Figure 1.

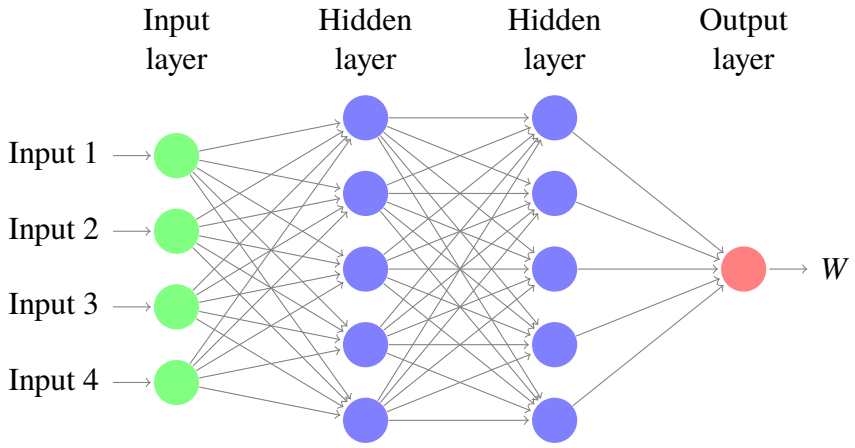

Figure 1: A deep artifical neural network, transforming some input V to some vector space W

In the neural network, the links between node j from one layer to a node i in a forwarded layer are associated by a weight w_{ij} and a bias υ_i. The total incoming flow ξ_i to node i is proportional to the summation over the weighted outputs from all predecessor nodes, according to the formula:

$$\xi_i = \upsilon_i + \sum_j w_{ij} x_j \tag{2}$$

where j is the subset of all predecessor nodes to a node i. The output from a node denoted x_i is in turn given by the non-linear activation function in node i, applied over the node's input:

$$x_i = h(\xi_i). \tag{3}$$

During training, the network is given pairs of inputs and desired outputs. A loss function L is defined, which penalizes the model when the predictions are wrong, and rewards the model when predictions are correct. By adjusting the weights of the network, the network is able to learn the transformation from inputs to the desired output by minimizing the loss function L. In each step, the weights w_{ij} are updated according to the so-called backpropagation algorithm, where we compute the partial derivative of the loss L with the respect to the weights [7]:

$$w_{ij}^{k+1} = w_{ij}^{k} - \alpha\Big(\frac{dL}{dw_{ij}}\Big). \tag{4}$$

By incrementally updating the weights, the loss is expected to converge to a local minimum. In practice, this means the loss L will stabilize at some value, indicating we have finished training.

As can be seen in equation (4), the constant α regulates how aggressively we update our weights. If the learning rate is too small, it will take a long time for the model to converge. Conversely, if the introduced learning rate is too high, the model may fluctuate and fail to converge to a local minimum altogether [8] .

2.1.2 Nearest Neighbor Search

In nearest neighbor search, an unseen data point is classified using the most similar items to that data point. The similarity is often computed according to some distance function D. The distance is often defined as the Euclidean, Cosine or Manhattan distance between two points [9]. However, any metric providing some meaningful metric distance on the data could potentially be used.

When classifying a point $\bar{u}$ we are often interested in find the K elements from a set U, which minimizes a distance

$$D(\bar{u}, u_i), \forall u_i \in U, \tag{5}$$

implying an exhaustive search over the entire set U. If $|U|$ is small this may not be a problem, but if the subset is large, millions of arithmetic operations may be required. To decrease the computational burden, we can employ one of or a combination of the following two techniques:

- Approximate Nearest Neighbor, involving partitioning the data, so that search is only performed on some subset(s) of U, where we are the most likely to find the best u_i.

- Product, Vector or Scalar Quantization. The common objective of a quantizer is to compress the vectors via some function Q, while still preserving as much correct distance as possible using the quantized representations.

We won't cover these techniques in detail as it is not in the main scope of the thesis, but will offer a short explanation of scalar and product quantization. In product quantization we split the vectors into several subcomponents, and perform K-means clustering on each of those subcomponents. During quantization, each of the vectors' subspaces are replaced by the corresponding closest centroid [10]. In scalar quantization, a vector V is estimated with a quantizer so that $\hat{V} = Q(V)$. If the values in V can take on any real number, they are split into several bins and the mapped to integer instead [11].

2.2 Natural Language Processing

NLP is a research area dealing with the interaction between texts and computers, including the ways computers can understand and process human natural language speech or text. NLP has a wide area of practical applications, such as for chat bots, speech recognition, language translation and text similarity analysis [12]. When processing text, it is common to first tokenize (i.e dividing) the text into several *tokens*. The purpose is to produce separate linguistic units, that can be used for further processing [13]. A simple kind of tokenization scheme involves dividing the text on hyphens and whitespace. If we are given the sentence "Paris in mid-August", the corresponding linguistic units (i.e tokens) in this case are identified as $Paris, in, mid, August$. However, it's also possible to define other tokenization strategies, such as splitting on sub-words and/or underscores. For convenience, we will use tokens and words interchangeably for the remainder of this section. Since, they both represent distinct linguistic units in texts, there is no conceptual difference between tokens and words. Rather, we treat them as artifact of the tokenization scheme implemented by the NLP developer.

In NLP, so called Language Models is an umbrella term for models trying to estimate the probability for a given sequence of words or tokens. In the following to subsections, we will present Statistical Language Models, and an extension to these, namely Neural Language Models.

2.2.1 Statistical Language Models

Statistical Language Models consist of a family of probabilistic models used to estimate the joint probability of a sequence of words $W = (w_1, w_2, ..., w_n)$. Formally, this joint probability is formally defined as

$$p(w_1, w_2, ..., w_n) = p(w_1) * p(w_2|w_1) * ... * p(w_3|w_2, w_1) * p(w_n|w_1...w_{n-1}), \quad (6)$$

where the probability of observing a word is conditioned on the previous word in the sequence [14]. This suggests that if we have observed the word "Barack" we are also relatively likely to observe "Obama". Statistical language models have direct applications in areas such as speech recognition, optimal character recognition and spelling correction,

where the context would imply that some words are more or less likely than others.

Computing Equation (6) may be not feasible for very long sequences W, so a common approach is to using a window size K. Instead of conditioning w_i on all of its previous words from the sequence, we condition it on the K previous words only. To train the model, a common approach is to iterate over a training corpus, and estimate the conditional probability based on the observed number of occurrences of a particular combination (of length K). We denote the count C and estimate the conditional probability by [14]

$$p(w_i|w_{i-1}, ...w_{i-K}) = \frac{C(w_{i-K}, ..., w_i)}{C(w_{i-K}, ..., w_{i-1})}. \tag{7}$$

While this greatly reduces the number of computations, there is still one question left unanswered. If we estimate the probability from the number of occurrences, what happens to new, unseen combinations during prediction? Consider the two sentences `"the cat was bicycling"` versus `"the cat was cycling"`. If the vocabulary only contains instances of the combination "was bicycling", we would assign zero to the probability of "was cycling" (and even do division by zero), since we have clearly observed that

$$C(was, cycling) = 0.$$

Finally, statistical language models may run into the so called *curse of dimensionality*. Let V denote the vocabulary from a training corpus - that is, a list of all unique words observed in in the corpus. According to [15], the number of free parameters to compute the joint probability for a sequence W are given by

$$l = |V|^K. \tag{8}$$

2.2.2 Neural Language Models

An extension of the statistical language models are the Neural Language models. They improve performance while reducing the dimensionality of the problem, by using distributed representations of texts. The neural models simultaneously learn an effective **feature mapping of words** alongside the **parameters of the joint probability**, according to the following function composition:

$$f(i, w_{t-1}, ..., w_{t-n+1}) = g(i, C(w_{t-1}), ..., C(w_{t-n+1})) \tag{9}$$

Here, we denote the feature vector mapping C, and the parameters of function of the joint probability distribution g. Formally, we choose to learn some parameters $\Theta = (C, \omega)$, where ω are the set of weights from the function g. The composition is defined as follows [15]:

- Using a lookup table with shape $|V| \times d$, C takes a word i for any word in the vocabulary V, and outputs a fixed-size real vector, such that $C(i) \in \mathbb{R}^d$.
- The function g is often implemented as a deep neural network. A common approach is to represent the input of the network with the concatenation of K encoded vectors

$$x = (C(w_{t-1}), C(w_{t-2}), \cdot, C(w_{t-K+1})). \tag{10}$$

The composition is illustrated in Figure 2. Note that the inputs are in practice vectors valued, and not scalars.

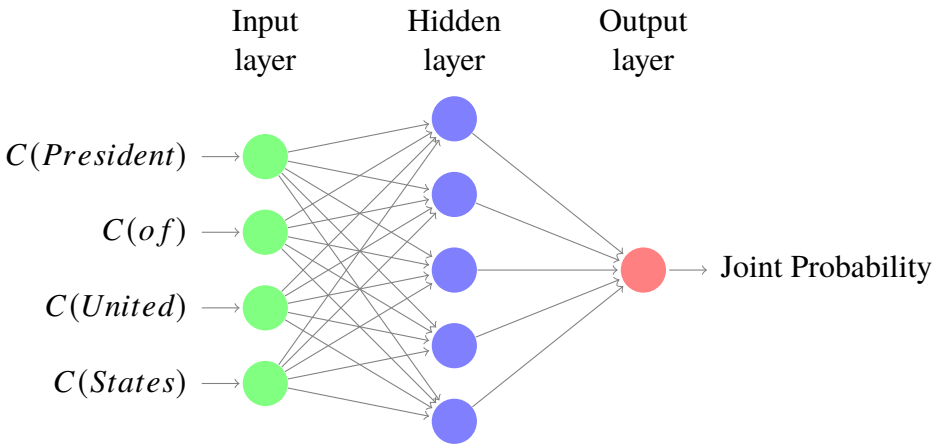

Figure 2: A neural network computing the joint probability of a an encoded sequence $w = \{C(President), C(of), C(United), C(States)\}$

During training, we can choose maximize the average log probability over the corpus, by iterating over all training sentences as one long sequence T

$$L = \frac{1}{T} \sum_{i}^{I} \log f(w_i, w_{i-1}, ...w_{i-n+1}; \Theta), \tag{11}$$

where Θ are the parameters of the neural network [15]. Importantly, the distributed representations generated from C are not only useful to estimate the joint probability of a sequence of words, but also for a various of downstream tasks. For instance, they could used to classify, cluster and measure similarity between words and sentences, as vectors between words occurring in a similar contexts will usually have a high correlation with each other, and/or have a low euclidean distance between themselves [16].

2.3 Bidirectional Encoders from Transformers

Bidirectional Encoders from Transformers (abbreviated BERT) are a special type of Neural Statistical Language models, developed and open-sourced by Google. In NLP, an *encoder* usually describes some function able to transform text of variable length into to a fixed-length vector representation [17]. As opposed to the traditional Neural Language Models presented in Section 2.2.2, words or tokens do not derive their embedded representation from a global lookup-table in BERT. Instead, BERT uses an encoder with 12 stacked and independent layers (so called transformer blocks) to derive token embeddings, conditioned on the words surrounding it. To achieve this, each of the 12 trans-

former blocks consists of a feed-forward neural layer as well a self-attention mechanism. The output of a block is used as the input to the next transformer block.

When feeding a BERT model with a text sequence, the sequence is padded by two or three special characters to denote the start, the end and optionally the separation between two passages in BERT. For single passages, the text will be padded as follows:

```
[CLS] The Weather is great here [SEP]
```
(12)

If we are dealing with a double sequence - as in cases where we are interested in understanding the interaction between two passages - the text is padded as follows:

```
[CLS] What is the weather like here?  [SEP] The Weather is great here [SEP]
```

Ultimately, at the last transformer block level, we will obtain one token embedding for each of the token in the sequence. If we are interested in obtaining a sentence embeddings, we can either use the token which marks the start of the sentence ([CLS] token), or compute the average over all the encoded tokens from a sentence [18].

2.3.1 Transformer blocks

The *Transformer blocks* was originally proposed by Vaswani et. al. For illustrative purposes, let h_i describe a vectorized representation of the word w_i, such that all tokens in a sequence W has a corresponding vectorized component in H. In each of the 12 transformer blocks, the self-attention mechanism refines the embedded representations of tokens by mathematically transforming a vectorized representation of a token, conditioned on the surrounding tokens [2], as depicted in Figure 3.

[2]This is a very complex topic, but we would like to refer the reader to the following tutorial, should they interested in understanding the self-attention mechanism in detail: https://nlp.seas.harvard.edu/2018/04/03/attention.html

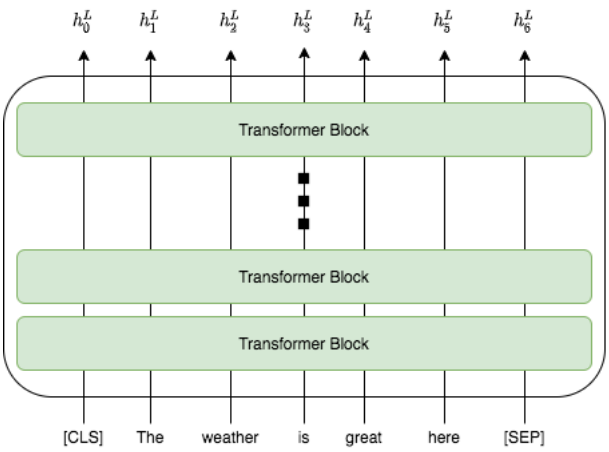

Figure 3: Visualization of the 12 transformer blocks in BERT

The transformation is performed via some learnt filter at that particular layer (a filter shared by all token embeddings in H) to produce three vectors each - denoted q_i, k_i and v_i [19]. Using matrix multiplication, the attention for all the tokens h in a sequence H are vectorized as

$$Attention(Q, K, V) = softmax\frac{(QK^T)}{\sqrt{d_k}}V, \tag{13}$$

meaning we output one "attention head" for each token in the sequence [20].

The important takeaway here is that the attention mechanism directs attention to other words in the sequence are "important" when deriving the vectorized representation for each of the tokens in sequence.

2.3.2 Practical considerations

BERT uses a wordpiece vocabulary, breaking down longer words into several, shorter subwords. This means BERT does not need to hold every possible word in a dictionary, and is able to create word embeddings even for misspelled words. This kind of extrapolation avoid detrimental situations where unknown words or variants of words would not get a representation, simply because they do not exists in the vocabulary.

BERT has a total number of 340 million parameters, so it is notoriously computationally expensive to train from scratch. To train a BERT model and all of its 12 transformer blocks, it employs something known as Masked Language Modelling and Next Sentence

Prediction. During training, it asked to restore randomly masked words, as in the following case

```
I went to the [MASK] to buy some groceries,
```

and to predict the most likely next sentence.

Under many cirumstances, even inferencing is slow, due to the number of parameters of the model. Still, BERT has gained a lot of popularity due to its transferability of learning to new tasks. Typically, one can fine-tune a BERT model on a relatively small, task specific datasets, by initializing the weights from publicly available models pre-trained on large document collections (such as all articles on Wikipedia). While pre-training the model takes a very long time, fine-tuning the model can be done in a few hours on a standard GPU [18].

2.4 Citation Recommendation System

In many online applications and databases, the true amount of available data may be overwhelming and difficult for the human to overview. A recommendation system is usually implemented to prevent information overload for a human, by presenting the most relevant subset of information. Most of us encounter recommendation systems on a daily basis, as they have been applied in a wide range of domains, such as movie or music suggestions on streaming platforms, targeted marketing in the e-commerce industry,as well personalized search engine results. Typically, the recommender makes suggestions based on what users with similar behaviour have liked in the past [21]. However, recommenders may also use *item-based* algorithms, to provide recommendation based on the direct similarity between items [22]. Intuitively, this translates to a situation where a user would be suggested an item X, if X is sufficiently similar to an item Y already liked by the user.

An application of recommendation system is the suggestion of scientific articles, known as citation recommendation systems. In response to some text written by the user, the goal of the citation recommender is to use this text passage - possibly combined with some metadata or global information - to suggest relevant scientific litterature [5].

2.4.1 TF-IDF for recommendation system

TF-IDF stands for term frequency inverted document frequency. It is an old but widely used statistical property within the fields of information retrieval, textual recommendation systems, or any application where dealing with similarity or relevance between texts is of interest. TF-IDF relies on a heuristic assuming terms occurring in relatively few documents carry more informational value, than terms which occur over many documents. Formally, the TF-IDF score s_j of a document d_j is expressed as:

$$s_j = tf(t_i, d_j) * idf(t_i), \tag{14}$$

where $idf(t_i)$ is commonly defined as

$$idf(t_i) = log(N/n_i + 1), \tag{15}$$

and $tf(t_i, d_j)$ the number of times t_i occurs in d_j. Here, N is the total number of documents in the collection and n_i the number of documents in which query token t_i occur. In practical settings, we often have to stem words to their root form, and use various techniques to correct the spelling, in order to account for syntactic differences in the text [23].

An explored and simple approach in citation recommendation systems, is to use the cosine distance between the query documents d and the citation document c_t based on their TF-IDF vectors [24]. Though it has been proved that TF-IDF often works very well in practice and is computationally very effective if implemented as an inverted index, TF-IDF suffers from several weaknesses. First, it fails to capture any meaningful semantics of a query, as TD-IDF does not consider word order, synonyms, negations or co-occurrence [25]. Secondly, whenever there is lexical gap between the query and the documents, TF-IDF does not infer the true relevance and importance of words [26]. Finally, it is not obvious why similarity according to TF-IDF necessarily implies relevance. Consider the following, somewhat extreme and perhaps unrealistic, example: If a user were to write about a comparative case study of the hotel chain Hilton's revenue management strategy in Paris against Los Angeles, a document about Paris Hiltons upbringing in Cal-

ifornia would certainly not be relevant, even though they may be considered very similar according to TF-IDF.

2.4.2 Probabilistic models for recommendations

In a probabilistic settings, the goal is to find the best citation c_t conditionted on a document d_t according to $p(c_t|d_t)$. To estimate this probability, we are required to bridge a semantic gap between the textual content in c_t to d_t. This requires a deep contextual awareness of the interplay between c_t and d_t.

One approach is to train the neural statistical language model to learn these probabilities. If we are interested in solving $p(c_t|d_t)$, we can - quite naively - assume that the words in the c_t are drawn independently, and estimate this function as

$$p(c_t|d_t) = \prod_{i=a}^{b} p(w_{t_i}|d_t), \tag{16}$$

Here, we let w_{t_i} denote the i:th word in the document t. If we use a neural probabilistic model, we can train a scoring function over some projected vectors $s_\Theta(w, d) = f(\cdot)$, where Θ are some parameters learnt during training of statistical language model to project words down into a vector space of arbitrary dimension. The probabilities for a word are now given by

$$p_\theta(w|d) = \frac{\exp(s_\Theta(w, d))}{\sum \exp(s_\Theta(w_i, d))}, \tag{17}$$

Finally, by substituting $p(w_{t_i}|d_t)$ with 17, we can obtain the most probable citations for a document [27].

In reality, words generated from human language are seldom (if ever) independently distributed. Also, words may derive their meaning from an interplay with other words. Clearly, the language can not be decoupled from its underlying context to be properly understood [28]. "Kentucky" has one meaning if we speak about the state, and another one if we talk about fried chicken. Similarly, the same word may have different importance in different contexts. If we say "Corona virus", we can leave "virus" out and still

preserve the overall semantics. Outside the medical context, "Corona" may suddenly get a completely different meaning, especially if we are placing an order in a bar.

2.4.3 Nearest neighbors with contextual embeddings for recommendations

Similar to the discussion in Section 2.3, we can provide recommendations using contextualized instead of global text embeddings. From the perspective of citation recommendations, we can learn a function to project query documents d with relevant citations c close to each other in a vector space. To get the most similar documents, we identify the ones with the lowest cosine distance to the query document, using a combination of textual or other features, known as nearest neighbors search [29]. The cosine distance between two vectors A and B is computed as the angle between the two [30, 31] :

$$cos(A, B) = \frac{A \cdot B}{||A||||B||}, \tag{18}$$

Learning the similarity between documents consists of an infinite number of combinations, so we can not train on all examples. Theoretically, it would be possible to train a multi-label classifier, but this would not generalize to unseen classes and would require many instances of each class. A common approach to estimate similarity between inputs is instead the usage of Twin neural networks. Twin neural networks are as the name implies two neural network sharing the same weights, trained in tandem. The two subnetworks forward one input each. The loss is defined a function of the absolute distance d of the feature vectors produced at the last layer, denoted $h_{i,L}$

$$d = |h_{1,L} - h_{2,L}|, \tag{19}$$

Intuitively, if the inputs are similar to each other, the loss function stipulates that they should be generated as such that the absolute distance between the two are minimized for positive examples, and maximized for negative examples [32].

2.4.4 BERT for citations recommendations

Various approaches of using BERT for citations recommendations (and conceptually similar problems) have been explored in the literature. Below, we present some of these.

- In [33], the authors treated the problem of recommending scientific papaers as multi-class classification problem, using a feed-forward neural network. As input, the authors produced the vectorized BERT representations of a text passage. The probabilities for a citation was then obtained by applying the softmax function on the network's output layer. The K most relevant documents were thus given by the K documents with the highest associated probabilities. This assumes multiple training instances for each document, and would not generalize to new documents.

- In [34], the authors explored BERT for a similar problem - the use of BERT in question answering and information retrieval applications. BERT supports binary sequence classification, where the input is the concatenation of the query text and document, separated by a [SEP] token:

  ```
  [CLS] who is the president in United states?  [SEP] Barack
  Obama [SEP]
  ```

  Using a this approach, a classification layer is added on top of the BERT representation, which computes the probability of this being a negative instances versus a positive instances. In other words, this suggests whether the retrieved document is relevant to the first statement or would help in answering that question. Also known as an interaction-based ranker, this considers the interaction between the two statements [35]. This however comes at a huge computational cost, as the number of inferences scales linearly with the number of documents in the collection. Even on a modern GPU, calculating the embeddings for hundreds of thousands up to a million of text passages would take 10's of minutes up to a few hours [36]. BING search engine uses more 2000 virtual machines equipped with GPU's to process 1 000 000 queries per second[3], displaying the computational effors required to put this into a production environment.

- Finally, the authors in [37] proposed Sentence-BERT - a framework for training BERT models using Twin neural networks. The authors had mostly single sentences in mind, such as estimating the similarity between "The airplane took of from the runway" and "The plane just lifted off the ground". With inspiration from

[3] https://azure.microsoft.com/en-us/blog/bing-delivers-its-largest-improvement-in-search-experience-using-azure-gpus/

computer vision, single sentences are project into a euclidean space, so that the distances between documents are comparable using cosine distance. [38]. The training procedure involves labeled sentence pairs, so that the model learn some transformation to project similar or relevant texts close together.

2.5 Graph algorithms

Graph algorithms is an important part of network analysis. In a network, each node is connected through a set of links to other nodes in that network to each other. Networks are assumed to express transitivity, meaning that if node A is connected to both B and C, then this increases the possibility of a connection between node B and C as well [39]. In academic literature, one type of network data is citation graphs, where each node is an article and each link between two articles is indicative of a citation. These type of networks are able to quantify similarity between two articles, that can complement or mimic text-based similarity. The assumptions is that groups of articles citing each other are considered similar to each other [40].

One type of graph algorithms is the node2vec algorithm, which takes a bidirectional and unweighted edge pairs as input and produces continuous embeddings for all of the nodes in the network. These continuous embeddings are for instance able to express similarity between the nodes via cosine distance. Producing these embeddings can be formulated as a maximum likelihood optimization problems, where the objective is to maximize the probability of observing a node's *neighborhood network*, given its feature representation. Formally this objective is defined as

$$P(N(u)|f(u)), \tag{20}$$

where $N(u)$ is defined as the neighborhood network of node u and $f(u)$ defined as the feature representation of node u. The neighborhood network $N(u)$ is sampled using a random walk from node u with maximum length l [41].

2.6 Metrics and loss functions

In this section, we will describe some useful metrics and loss functions, used when training our model and for performance evaluation on a different set of tasks.

2.6.1 Spearman rank correlation

The Spearman rank correlation describes how likely it is that a set X can predict a set Y. It is then defined as

$$r_s = 1 - \frac{6 \sum d_i^2}{N(N^2 - 1)}, \tag{21}$$

where d_i is the difference in ranks between the ranked variables $rg(X_i)$ and $rg(Y_i)$, and N the total number of the samples. The Spearman correlation captures any monotonic relationship between the sample, meaning that if sample X increases, then Y either decreases or increases monotonously with the rate of change of X [42].

2.6.2 Normalized Discounted Cumulative Gain (NDCG)

Normalized Discounted Cumulative Gain is a metric used to measure the performance of some ranking function y_r^f of a dataset S_n. In information retrieval, n is usually defined as the top K elements . The Cumulative Gain (CG) is defined as,

$$\sum_{r=1}^{n} y_r^f, \tag{22}$$

while the *discounted* variant is by given by

$$\sum_{r=1}^{n} y_r^f D(r), \tag{23}$$

The factor $D(r)$ takes the ordering into account by decaying Cumulative Gain according to

$$D(r) = \frac{1}{log(1 + r)}. \tag{24}$$

Finally, Normalized discounted cumulative gain (NDCG) is is obtained by diving DCG

by the ideal or maximum DCG: is defined as:

$$NDCG(f, S_n) = \frac{DCG(f, S_n)}{IDCG(S_n)}.$$

3 Data

As part of building our recommendation system, we primarily work with three kinds of data:

1. Scientific papers in diagnostic imaging
2. An open access image database in biomedicine, with notes of image findings
3. Online conversations between doctors on a digital platform concerning clinical findings.

We devote the three sections below to describe the attributes of these three datasets and how they were obtained.

3.1 Scientific papers in diagnostic imaging

PubMed is a scientific database for biomedical litterature, maintained by the american National Library of Medicine. To date, PubMed has more than 30 million scientific papers (as either full-text or abstract only) from a wide range of biomedical topics, such as radiology, biomolecular engineering, and physiology. PubMed was selected after consultation with several radiologists, suggesting it was probably the biggest database collecting scientific papers in medicine and a common "go-to" database when searching for papers. More the point, PubMed offers several well maintained API and ftp severs for bulk download, which was advantageous for data processing reasons.

A scientific paper on PubMed usually consist of a title, an abstract, authors, keywords and a set of other attributes such as citation count and journal. Often, they contain a link to the full-text paper should someone want to read it. For practical reasons, we selected a subset of articles from PubMed we think are relevant in diagnostic imaging with the following boolean query:

```
"radioscopy OR fluoroscopy OR radiology OR radiographics OR radiography OR
  radiological OR mri OR ct OR computed tomography OR magnetic resonance
                    imaging OR x-ray OR imaging"
```

From this subset, we selected only the articles that also had been manually labelled with the keywords "diagnostic imaging" or ("diagnosis" and "radiology"), yielding a total of about 379 000 articles. This subset selection was done in consultation with radiologists having a research background, but we do not suggest this is the only or necessarily the best way. One of its biggest disadvantages is that newly published papers (up to a few months) on PubMed seldomly have been manually tagged and categorized by a human, which means that this will necessarily decrease recall. A typical example of a PubMed article is displayed below:

> **Title**: "MRI of anal canal: normal anatomy, imaging protocol, and perianal fistulas: Part 1."
>
> **Abstract**: "Anal complaints are very common in the general population and are caused by a variety of disorders mostly benign in nature. The aim of this article is to provide the radiologist with a detailed description of the MRI anatomy and technique, and an overview of the various diseases most commonly presenting with anal pain, by descriptions and illustrative examples of MRI features of each entity."
>
> **Pmid** 24223633

3.2 Biomedical image search engine

The *Open Access Biomedical Image Search Engine* is a database maintained National Library of Medicine in USA[4] - the same as the maintainer of PubMed. Of interest to us, it contains several hundred of thousands references to open access full-text articles, which contains detailed descriptions on what the radiologists would expect to see on an image related to a particular disease, syndrome or health condition. For example, PubMed article with pmid 24223633 is an article about of diagnostic imaging of *aneurysmal bone cyst*, and as part of the article the authors has provide a set of reference images with detailed description of the relevant findings in them:

[4] https://openi.nlm.nih.gov/

Keywords: "spinal tumor; aneurysmal bone cyst; tumor of the bone"

Image findings: "63-year-old female patient with an expansive, lytic and unicameral cyst lesion located in the right appendix of T7 (arrows). (A and B) CT scan and T2-weighted MRI imaging prior to surgery revealed an expansive, lytic and unicameral cyst lesion located at T7. (C and D) Sagittal fat-restrained and T1-weighted MRI revealed a highly hydrated lesion located at T7. (E and F) Anterior-posterior and lateral view X-ray post-surgery. Segmental instrumented posterior fusion from T6–T8 was performed after the tumor was removed. CT, computed tomography; MRI, magnetic resonance imaging."

Pmid 24223633

We selected a subset of these by filtering on the following terms to be included in either the keywords or image findings:

```
mri OR CT magnetic resonance imaging OR imaging
```

and where the article itself satisfied the criteria of being manually tagged as "diagnostic imaging" or ("diagnosis" and "radiology"). The final subset now contained 29808 description of image findings, and a reference to the article where they were written in. We will later show how we used these to fine-tune a statistical language model, linking description of image findings to scientific papers.

3.3 Case data

A case is a discussion about a specific patient with at least one post on the platform. The discussions carry the following information:

- The title of the case, chosen by the case author/original poster
- One or more blobs of unstructured free-text posts from radiologists, found as both the original post and replies
- In the blobs, radiologists are free to write whatever they want about the patient case (or anything else, should they want, since it's an online forum), meaning it can be of varying informational value. It is mandatory to write at least some kind

of free text in the original post as well as in any of the thread replies.

- Two optional field - suggested diagnosis and suggested-follow up. These can be found both in the original post as well as in the thread replies.
- The medical sub-speciality area
- Finally we have the medical meta-data, commonly referred to as DICOM data. It contains information about what radiographical modality was used (MRI, CT, x-ray, ultrasound etc), what body part was examined and some more information.

From the platform, we collect 35 labelled cases as development set, and 16 for gold standard validation. A label of a case discussion contains a set of unique identifiers resolving to a scientific articles from the medical database PubMed. A full example is provided in Appendix A. Free-text written by doctors for clinical purposes is typically informationally dense. The language is often domain specific with jargon, abbreviations and acronyms. Additionally, the text usually contains linguistic and grammatical error, lacking complete sentences and auxiliary verbs (be, is, are). Clinical notes are often misspelled [43]. Because the radiologists from the platform do not necessarily have English as native language, we believe there are particularly important considerations.

4 Method

Based on our problem definition and our two envisioned core functionalities of the system - that is, recommendations of *case-to-articles* and *article-to-articles* - we formalize two objectives. First, in the case-to-articles functionality, we are interested in learning the following scoring function, mapping a scientific paper and a case to a relevance score

$$f(y_j, X) = S_j. \tag{25}$$

In the above equation, we let X denote a set of scientific papers and y_j a medical case. The output is a vector valued relevance score S_j in the range of $\in [0, 1]$, where 1 would indicate perfect relevance and 0 would indicate perfect irrelevance.

Secondly, we believe the overall user experience and training of models will first benefit from knowing what articles are similar to each other, motivating the *articles-to-articles* functionality. Intuitively, if we know that article x_1 is of interest to case y_j and x_1 is similar to x_2, it is probable that x_2 is also of interest to y_j. As such, we also try to learn the following function

$$g(x_j, X) = S_j, \tag{26}$$

To build these two functionalities, we consider a semi-supervised workflow, essentially generating two different subtasks (one for case-to-articles, and one for articles-to-articles). We employ the following high-level strategy to build our two core functionalities:

1. In articles-to-articles, we first draw inspiration from Bhagavatula et al (2018), stating citation networks can be used as a proxy for similarity between scientific articles. Since we lack explicit labels for similarity between article pairs, we build a citation network between scientific papers and project them into a euclidean space, so that the distances between two vectors can be measured with cosine or euclidean distance [29]. We use an implementation of the node2vec algorithm to generate graph embeddings for scientific papers.

   To actually approximate our similarity function $g(x_j, X)$, we fine-tune a pre-trained BERT model to produce text embeddings for scientific papers, so that the distances between two vectors can be measured with cosine distance [44]. We call this encoding function C_1. We perform unsupervised labelling by assigning the similarity scores as a function of the proximities in the citation network, as measured by the cosine distance between their generated graph embeddings. The ultimate goal is to be able to recommend similar to articles as one given by the user.

2. In case-to-articles, we explore to what extent we can use produced text embeddings for the retrieval of scientific papers based on notes from a patient case. Ideally, our goal is to project topically similar cases and articles close to each other in the euclidean space. We construct a new encoder, and denote this encoder C_2. To build and evaluate this particular encoder, we use three different datasets:

- A training set, consisting of roughly 40 000 data points from the *Open Access Biomedical Image Search Engine*.
- A development set, consisting of 35 cases from the platform, for which we report the **Precision** on.
- Finally, a human assessed validation set of 16 cases for which we report the **NDCG** on, which we call the gold standard dataset. We think this is a highly unbiased way of evaluating the true "accuracy" or usability of the recommendation system.

Before we go into details on each of these, we highlight some of their differences and similarities in the figure 4. Importantly, different datasets are used and the fine-tuning is performed on different pre-trained models[5]. There is also some slight differences in the assessments of the two models.

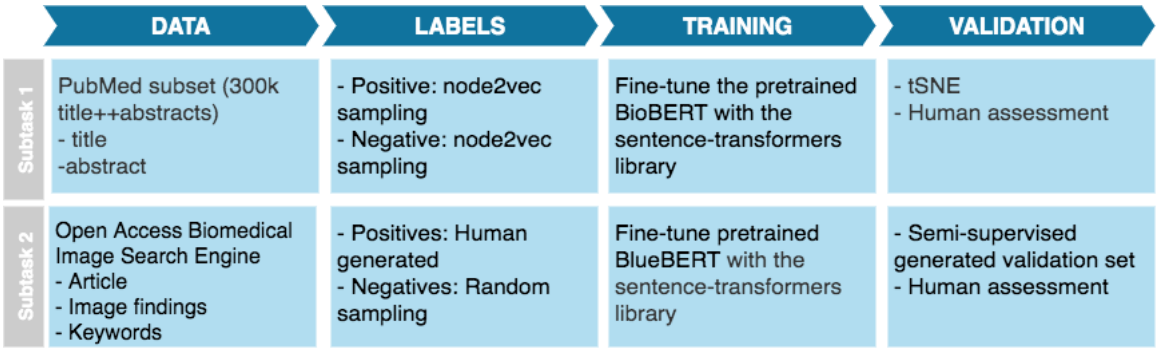

Figure 4: The workflow for Subtask 1 (articles-to-articles) and 2 (case-to-articles)

4.1 Articles-to-articles

In this subsection, we will briefly explain how we build the *articles-to-articles* encoder, including the semi-supervised generation of a dataset using graph embeddings.

4.1.1 Citation network

In academic settings, the citation network can be used to estimate similarity between scientific papers. It is often considered that papers citing each other are somehow related

[5]It is worth pointing out that the choice of using two different encoders are merely an artifact of how we initially approached these two problems, rather than us suggesting that two different models are absolutely necessary. In fact, it may have been possible to merge the two training procedures together, to create one single model.

to one another [45]. A citation network don't generalize trivially to unseen documents however, and the model has to be aware of any changes in the underlying structure as the network grows or changes to reflect the current state. We assume however that we can use a static snapshot of the citation graph to construct gold standard similarity labels between the scientific papers in our subset to be used during the fine-tuning of our BERT model. The main purpose of the graph embeddings is thus to generate a labelled dataset that we can train our statistical language model on.

We collect all the citations between articles inside of our subset by using a publicly available PubMed API[6], including both incoming citations (cited-by-others) as well as the outgoing citations (references-to-others). We then represent this network as an undirected edgelist as following:

$$20658266, 20118717$$

$$20118717, 22526116$$

This yields a total of 4 900 000 links generated from the 300 000 abstracts. Arguably, citations are not truly bidirectional, but for simplicity we make the assumption that we can treat these connection as "non-directed".

We train our graph model using the previously proposed node2vec[7] to compute the graph embeddings. The exact used hyperparameters are outlined below in Table 1. We performed no experimentation of these hyperparameters. Instead, we use the default ones provided by the library, and discovered that they were sufficient to provide good results.

Walk Length	**Epochs**	**Return Weight**	**Neighbor Weight**	**Dimensionality**
10	20	1	1	128

Table 1: Hyperparameters for node2vec the algorithm

[6]https://www.ncbi.nlm.nih.gov/home/develop/api/
[7]https://github.com/apierleoni/graph2vec

4.1.2 Sampling training pairs

With embeddings for more than 300 000 citations, we also have equal number of permutations of possible training pairs. This is obviously not possible to train on in finite space and time, so we index all the embeddings using Faiss [46], which is a library with Python bindings for nearest neighbor search in high dimensions. We can then obtain the one nearest neighbor for each of the scientific paper in our subset, from which we randomly sample 2 negative scientific papers whose distance is less than 0.4. It was assumed anything bellow 0.1 in cosine distance is equivavlent to "being similar" and anything above 0.6 as "being dissimilar", so that the fine-tuning could be formulated as a binary classification problem. However, one could also train on the gold standard with mulitple labels (i.e discretize the distances into several bins) or on the cosine distances directly.

We now have a total of roughly 800 000 training pairs. For practical reasons, we only train and validate on a balanced 10% of these, as this is still a substantial amount of data.

4.1.3 Objective

Recall our objective defined in equation (26); that is, given two articles, we are interested in assigning a score depending on how similar these two are to each other. Consequently, to find the most relevant article to an article given by the user, we condition equation (26) on the given article x_i and take the largest score with respect to all other articles X. If we can train an encoder C_1 to project similar articles to an article x_i in a euclidean space by their embedded representations, the most similar article can be found by returning the K articles with the smallest cosine distances from x_i

$$\underset{i \in I}{\operatorname{argmin}} \quad \left(\frac{C_1(x_i) \cdot C_1(X)}{\|C_1(x_i)\| \|C_1(X)\|} \right). \tag{27}$$

If we consider two text passages below, we can see the problem entails far more difficulties than simply just determining the number of overlapping terms:

```
s1 = "The validity of hyperdense lumen sign in non-contrast
chest CT scans in the detection of pulmonary thromboembolism.
It is possible to identif...."
```

```
s2 = "Unenhanced multidetector computed tomography findings in
acute central pulmonary embolism..."
```

CT is short for *computed tomography*, and in many contexts, *pulmonary thromboembolism* refers to the same things *pulmonary embolism*. Even though the above passages share few words in common, they are topically very similar, so we would want our embeddings to encapsulate these kind of relations, even though the only word in common is *pulmonary*.

4.1.4 Training

We fine-tune our encoder by initializing the weights of the network from a publicly available pre-trained model called BioBERT. BioBERT is pre-trained on predicting masked words and next sentences from over 270 000 full-text articles and 200 000 abstracts from the biomedical database PubMed, making it suitable for NLP tasks in biomedicine [47]. We implement the training using the Sentence-Transformer[8] package, which is an interface for training twin neural BERT networks in PyTorch to produce sentence embeddings for projection into the euclidean space. When training twin neural network, there are - as the name suggests - always two sub-networks sharing the same weights, as seen in Figure 5.

[8]Though the package is called sentence-transformer, the concept of "sentence" in BERT is different from the concept of a sentence in human text. A sentence in BERT may refer to a text passage or a paragraph as well. See [18] for a discussion on this subject

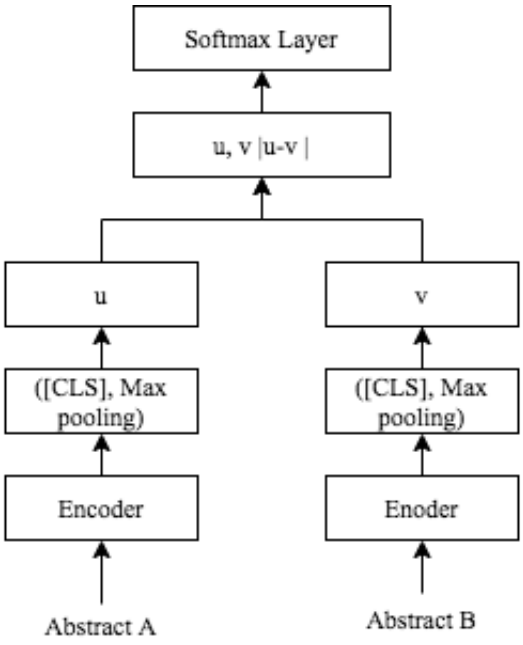

Figure 5: The architecture of a Twin neural network

BERT produces an embedding for each token in the text as well as an embedding for [CLS] token representing the entire sentence, meaning one must actively select a pooling strategy. We use a combined "Max" and "[CLS]" pooling strategy as the final representation. We speculate the [CLS] embedding entails an overall representative representation of the text, and that the embedding of the biggest magnitude will point in what direction the most important token of the text points at. Since the tokens derive their embeddings by their local context, we think this will be representative of the single most important topic in the text.

During prediction, only one network is used to produce the representation, so we can propagate two articles through the network and measure the cosine distance between the two obtained vectors.

4.1.5 Preprocessing

We perform some pre-processing, including the removal of non alphabetic letters, with some exceptions. We only remove numbers if the are surrounded by a word boundary, meaning we keep numbers for words such as "COVID-19", "T2", "C1" but remove them if in examples such as "21 patients was selected for this study". We remove all kind of punctuation, except for dashes, such as in the case of "T1-Weighted" and "CT-scan". Fi-

nally, we concatenate the title alongside abstract text, treating the title and abstract as a single united text passage.

4.2 Case-to-articles

4.2.1 Objective

Recall the objective for case-to-articles is to map related cases and scientific, with articles describing these image findings close to each other in euclidean space (an irrelvant ones far apart). If we can train a text encoder C_2 for this purpose, we can approximate the maximization of equation (25) - i.e the most relevant article $x_i, \forall x_i \in X$ in light a case y - with the cosine distance between the vectorized representations

$$\underset{i \in I}{\operatorname{argmin}} \quad \left(\frac{C_2(x_i) \cdot C_2(y)}{\|C_2(x_i)\| \|C_2(y)\|} \right). \tag{28}$$

As an example, we present a case and an associated relevant scientific article, talking about a patient affected by the Corona (or COVID-19) virus (we mask out some details that could potentially identify the individual with "XX"):

Case Description

```
Abdominal CT in XX year old patient with recent XX surgery. COVID
confirmed later.
Presents fever and abdominal pain.
Lung bases with GGO. A PCR test later on confirmed COVID neumonia.
Always look at the lung bases, and these days even more!
```

A Relevant Scientific paper

```
"Coronavirus (COVID-19) Outbreak:  What the Department of Radiology
Should Know.


In December 2019, a novel coronavirus (COVID-19) pneumonia emerged
in Wuhan, China.  Since then, this highly contagious COVID-19
has been spreading worldwide, with a rapid rise in the number
of deaths.  Novel COVID-19-infected pneumonia (NCIP) is characterized
by fever, fatigue, dry cough, and dyspnea.  A variety of chest
imaging features have been reported, similar to those found
in other types of coronavirus syndromes.  The purpose of the
present review is to briefly discuss the known epidemiology and
the imaging findings of coronavirus syndromes, with a focus on
the reported imaging findings of NCIP. Moreover, the authors
review precautions and safety measures for radiology department
personnel to manage patients with known or suspected NCIP. Implementation
of a robust plan in the radiology department is required to
prevent further transmission of the virus to patients and department
staff members."
```

If we use the two exmples above, our function $f(\cdot)$ should produce a high score for this combination, as these two are highly related to each other from a radiologist's perspective. Again, we cannot reply on the number of overlapping terms. For example, the doctor doesn't use the full name of the virus (COVID vs COVID-19) and has misspelled *pneumonia* as *neumonia*. Also, the authors of the scientific paper have decided not to use spacing between the word *corona* and *virus*, treating it a compounds word. This highlights how different writing styles affects the syntactics of the texts, while they semantically refer to the very same thing.

4.2.2 Training

The setup for Subtask 2 is essentially identical to Subtask 1, with the same Twin neural network architecture and pooling strategy (se figure 5 for a reference to its implementation). However, this time we use a different dataset to train on and a different pre-trained model called BlueBERT that we initialize our model weights from. BlueBERT is in turn initialized from BioBERT and then pre-trained on clinical notes from the MIMIC-III, a collection of de-identified clinical notes from 40.000 patient hospitalised at Beth Israel Deaconess Medical Center in Boston, USA [48].

We fine-tune the model on the *Open Access Biomedical Image Search Engine* subset, training siamese network on pairs of [image finding, title++abstract] with a corresponding binary label. For positive instances, we use the image findings and the corresponding article they were found in. For negative instances, we randomly sample them from within the collection, so that the dataset is balanced with 50% positive and 50% negative examples. Theoretically, a balanced dataset would prevent the model from always trying to correlate everything. Obviously, there are more advanced heuristics we could use to samples negatives instances, but we think the data is heterogenous enough for random samples to be sufficient. We use 70% of our examples for training, and 30% for test.

While *Open Access Biomedical Image Search Engine* differs from the clinical corpora on the platform, BERT has shown tremendous results on a wide range of tasks with little fine-tuning [18]. We believe if the underlying statistical language model has a notion of what clinical language looks like, but is asked to classify peer-reviewed text of image findings from academia on top of this, it will still generalize to instances where the text

is more unstructured from the clinical domain. To test this hypothesis, we collect a development set from the radiology platform, where we have a total of 35 cases labelled with 5-15 relevant articles each. By using the results from Articles-to-articles, we are able expand the number of relevant articles, so that we on average have about 200 relevant articles for each case, which improves the validity of the evaluation. It is assumed that any article that has a cosine distance of less than 0.07 to a user selected articles, should automatically be appended as a relevant article to that case. The number of relevant articles for each case are now given by the following distribution in Figure 6:

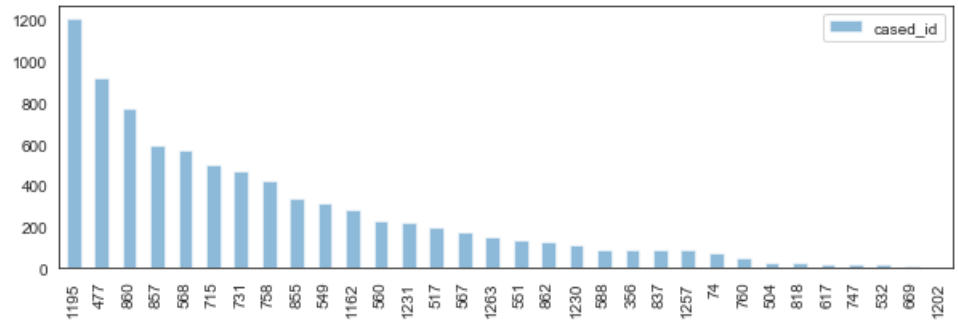

Figure 6: Number of positive examples per case

Finally, and most importantly, we select 16 unseen cases for validation data, and let our recommendation engine suggest 5 articles for each of these. It is then up to a human expert annotator to assign a relevance score of either 0, 1 or 2, depending on their subjective opinion. We use this for gold standard evaluation, as we think this most closely resembles the true performance of the recommendation system.

4.2.3 Preprocessing

We follow the same steps of preprocessing as in the previous subtask. In addition, we concatenate keywords and image findings *Open Access Biomedical Image Search Engine* and prepend a hashtag to all the keywords. The reason behind the hashtags is that doctors on the platform sometimes uses hashtags when they want to highlight extra important concepts while describing the image findings or the patient diagnosis, so we try to emulate this kind of writing style in the training data as well.

4.2.4 Prediction

During inference, we treat all posts as separate entities. We thus infer the most relevant scientific papers from a single post, and do not account for the accumulated knowledge in the post.

4.3 Use cases and Inferencing

Our framework allows us to ask two types of queries from the recommendation system.

- One, the user may feed the system with a scientific paper x_j it likes, and ask it to return similar ones back. Formally, we would retrieve the K documents with the lowest cosine distance according to equation (28).

- Two, we may feed the system with a clinical case and ask the system to recommend articles relevant to that clinical case. Again, we would first obtain a candidate set of topically relevant documents to case y by finding K first element with the lowest cosine distance.

The user could choose to get either the top K results back, or all the L articles with satisfy a certain similarity threshold. For instance, they could specify that they would want to see all articles whose cosine distance is less than 0.1, or some other arbitrary number. If the retrieved articles is hooked onto a database or an external service, it is also possible for filter for other metadata, such as year, publication type, authors or number of citations as well any other metadata that would be of relevance to user. A UX designer associated to the digital platform created a mockup of how these articles could be presented to the user, as visualized by the blue box of Figure 7.

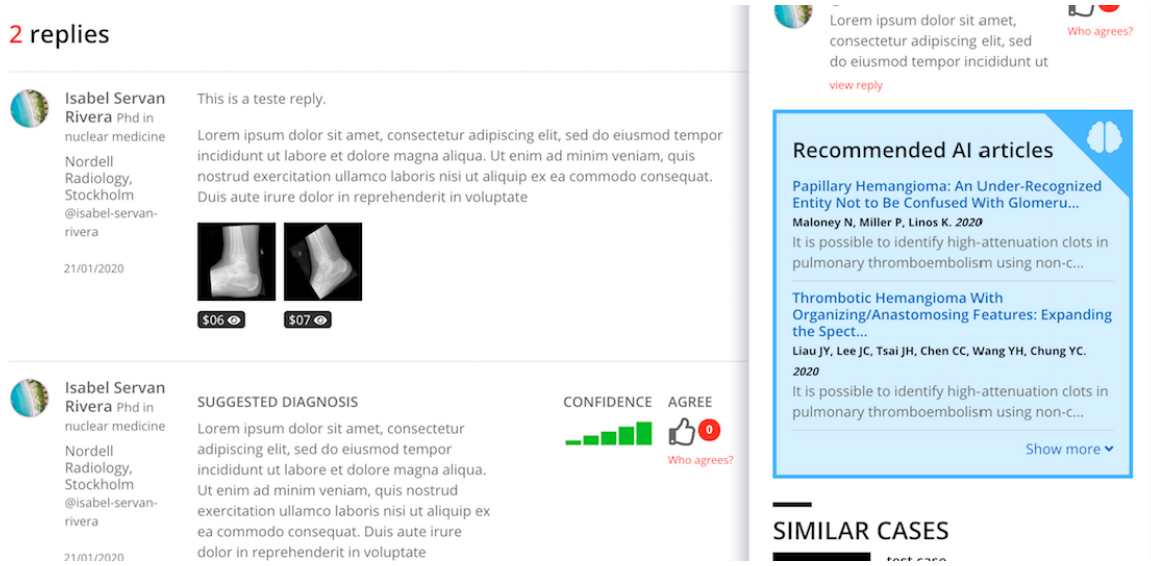

Figure 7: A mockup of how the recommended articles could be presented to the users on the platform.

4.4 Motivations

There are a few conceptual advantages with our two encoders. First, BERT has the ability to learn what words are important given the surrounding context are important, which will help in contextualized retrieval. It is suggested that BERT representations yields several subspaces, entailing several semantic meanings of a text passage [49]. Hopefully, our model will be able to encapsulate these subspaces to something comparable when encoding our text passages. Secondly, BERT uses a wordpiece vocabulary, so it splits words into chunks. This can help in misspelling, singular vs plural and inflected words (such as cancerous and cancer).

Lastly, propagating a text through the BERT network is computationally expensive. With our approach, we can precompute the encoded embeddings $C_1(x_i)$ and $C_2(x_i)$, $\forall x_i \in X$ and index them in a high-performance library like Faiss[46] for nearest neighbor search. This is required in scenarios where we can not afford to spend hours on inference.

There are two weaknesses however, which are worth pointing out for transparency. One, we do not know whether texts from different domains - case and scientific papers - can be projected into the same euclidean space. Secondly, when using nearest neighbors,

there is no straightforward way to measure the uncertainty in our prediction.

5 Results

5.1 Articles-to-articles

5.1.1 Graph embeddings

Generating graph embeddings from the node2vec algorithm is an unsupervised task, meaning we have no labels to measure our performance with. We can project the graph embeddings down to 64 dimensions using PCA, and then apply tSNE to visualize the clusters it has produced. We can inspect the results in Figure 8, and by the looks of it, there seems to be at least some level of separation between clusters. To generate this graph, we performed an unsupervised expansion of the the sets of relevant articles associated to each case, automatically adding new scientific papers with a cosine distance lower than 0.1. Both PCA and tSNE are available in the scikit library [9].

[9] https://scikit-learn.org/stable/

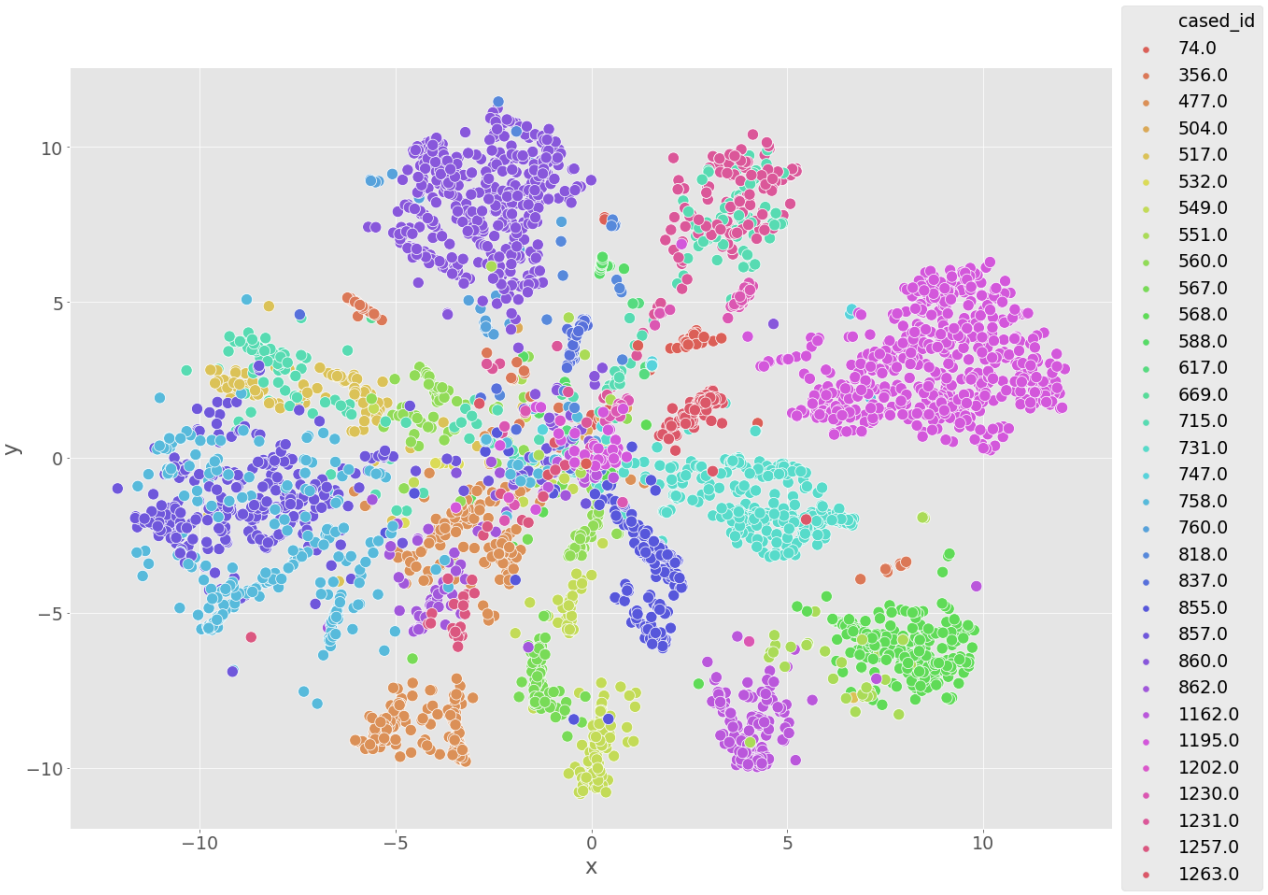

Figure 8: A tSNE projection of articles marked as relevant for some 30 different cases.

The tSNE projection offers several interesting interpretations. Possibly, the more specific an article is to a topic, the further away it is from origin. Papers with a broader perspectives in a topic end up closer to the origin, and are harder to distinguish from other scientific papers, because they are also cited by paper outside their own topic.

5.1.2 Encoder training

We fine-tune the BERT model for 6 epochs to produce word embeddings whose cosine distance between similar papers are minimized and maximized between dissimilar. We notice little improvements training it for anything beyond a few epochs and obtain a validation accuracy of 85.6% and 89.9% as measured by the Spearman and Pearson correlation respectively. This behvaiour can be obeserved in Figure 2, where the validation accuracy is plotted against the number of epochs:

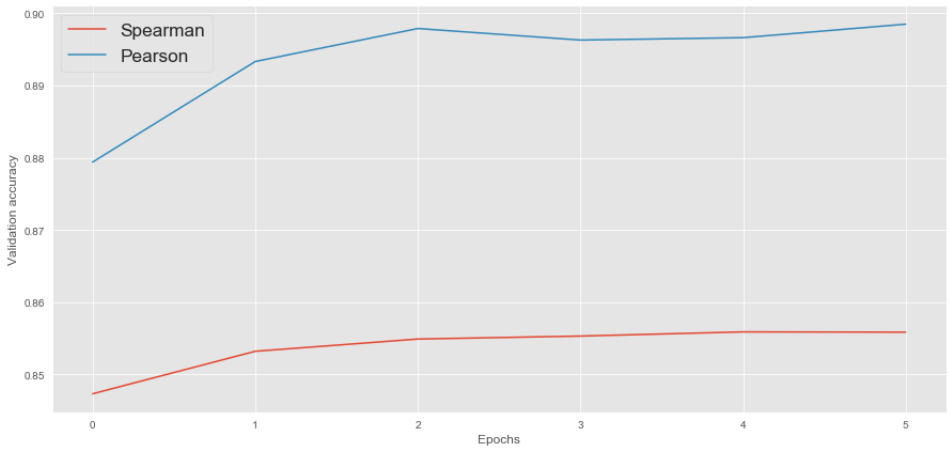

Figure 9: The validation accuracy with respect to the number of epochs.

Alongside the correlations, we conduct a qualitative assessment on 10 randomly samples articles and manually rank the output of the 5 suggested closest articles according to our BERT model. We ask a human annotator with domain expertise to rank either 0 (unrelated), 1 (somewhat similar) or 2 (very similar) on each of the sampled paper.

Pmid	Ranking quality					Suggested Top 5	NDCG
8284406	2	2	2	2	2	8372181, 15891888, 15732558, 8165977, 8165977	1.0
8205492	2	2	0	2	0	1863503, 8370830, 7991812, 11157403, 28500363	0.83
7822914	1	2	1	1	1	29143872, 22072191, 8982922, 12954266, 6643972	0.8
10845496	2	2	2	2	2	29476440, 12612475, 9648798, 7886471, 2255944	0.896
3830524	2	2	2	2	2	8774557, 30230762, 23994909 29952971, 7884486,	1.0
29058098	2	2	2	2	2	27190771, 28473089, 1299726, 17118291, 2798868	1.0
8685344	2	2	2	-	-	27190771, 28473089, 1299726,	0.83
19673029	2	2	2	2	2	17268711 12207186 22191287 2660295 22191288	1.0
16160117	2	2	2	2	2	27209201, 22365523, 29228461 19386100, 15116655	1.0
15100124	2	2	2	2	2	12438034, 9134582, 8431691 12395259, 10901925	1.0
20462991	2	2	2	2	2	23089877, 7696054, 8641070, 12773675, 8240470	1.0
Avg	1.95	2	1.7	1.7	1.5	23089877, 7696054, 8641070, 12773675, 8240470	0.96

Table 2: Human assessment of article to article recommendations

As indicative of the table 2, the human annotator's perceived judgement of our suggestions are quite high, showcasing the model's usefulness in displaying topically similar articles.

We also perform the unsupervised algorithm tSNE on a total of 6000 articles associated to our validation set, as seen in Figure 10. The cluster identified by the tSNE algorithm sugests the embeddings of the scientific papers produced by BERT *can* be projected down to the euclidean space. Here, we can observe a much clearer separation of the topics, compared with the graph embeddings in Figure 8.

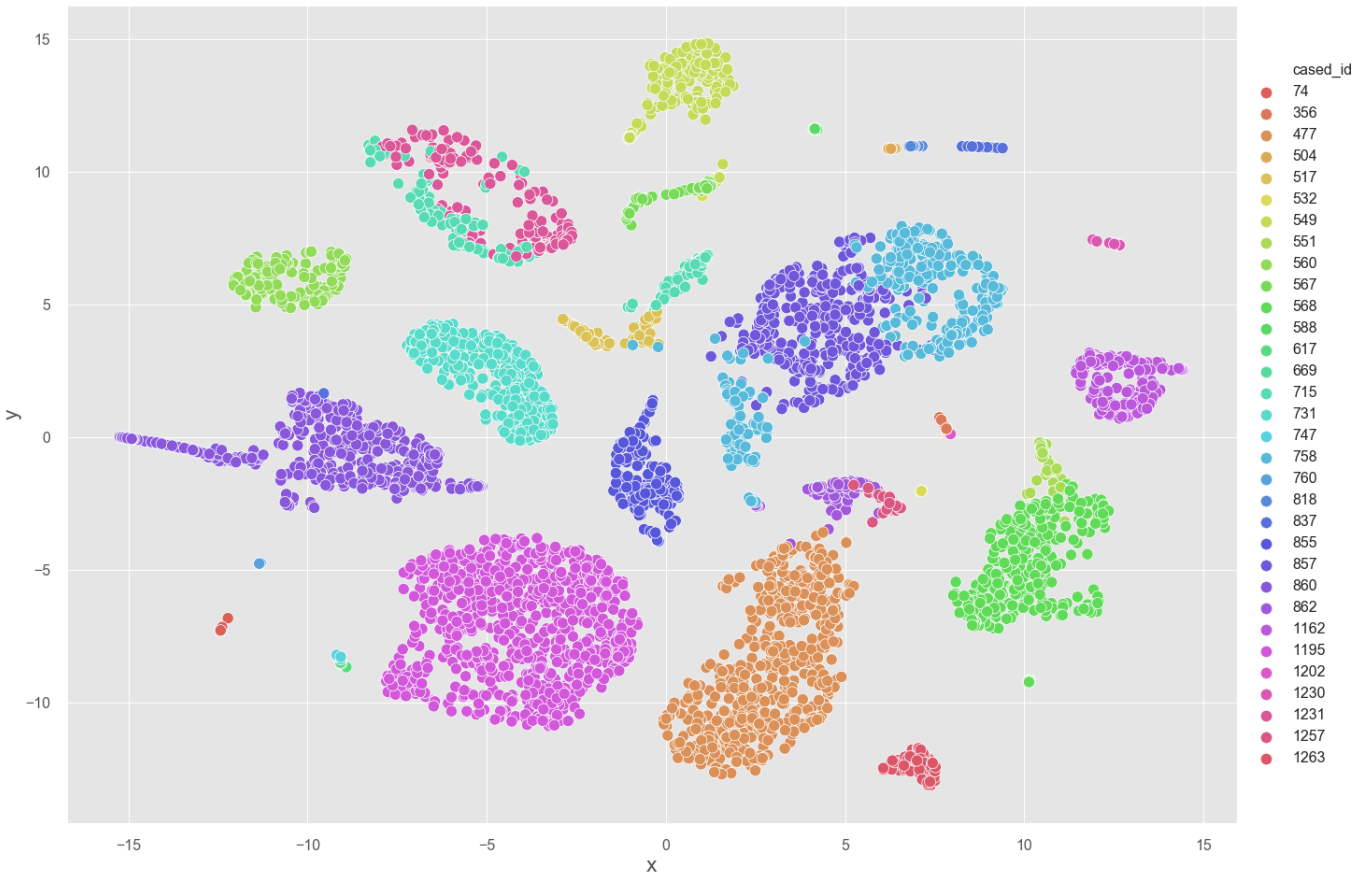

Figure 10: A tSNE projection of the suggested articles embeddings for some 30 cases.

5.2 Case-to-articles

We fine-tune our encoder C_2 for one single epoch, since it converges to a validation 92.5% accuracy on some of the held-out training data (Recall the training data consists of some 40 000 data points from the Open-I collection). We do not train the model for longer, primarily because we already have a large dataset and we do not want the model to fit too well to our training data. This may seem counter-intuitive, but we want to preserve as much of the underlying language understanding as possible from the model, since the training data differs somewhat from the data in the practical setting.

We measure the precision of our suggestions at three different thresholds: 3, 5 and 10 recommended articles. The results at the different thresholds on our development set are presented in Table 3, and there is support that between 3 and 4 out of 10 recommended articles are on average considered relevant

Furthermore, as dscribed in the method, we conduct a manual assessment of 16 articles

P@3	P@5	P@10
0.396	0.375	0.334

Table 3: Precision on validation set

on a gold standard validation set. We ask a human expert annotator to rank either 0 (unrelated), 1 (somewhat relevant) or 2 (very relevant) on each of the sampled paper. From these evaluations, we obtain a normalized discounted cumulative gain of 0.63 ± 0.3. Meanwhile, the precision - if no distinction between "somewhat relevant" and "very relevant" is made - is obtained at 0.54 ± 0.32.

5.3 Retrieval Performance

Depending on the exact setup, our predictions can be performed in a couple of 100ms up to a few seconds on a standard MacBook Air, with a 1,6 GHz Intel Core i5 and 8gb of memory. The best of out of five rounds suggests encoding a text passage of standard lengths (as we have do to do in Subtask 2) takes 0.33 seconds. The retrieval phase (which we have to do in both subtasks) takes less than 0.13 seconds on a hot index in cache, using 8-bit scalar quantization and partitioning with 4096 clusters and number of probes = 20.. Menawhile, the size of each index is 512mb for roughly 379 000 articles, and the raw PyTorch model size is about the same (474mb).

6 Discussion

This thesis offers several interesting implications. First, we have shown how statistical language models can achieve high precision on text-based recommendation tasks while using precomputed embeddings, even for longer text segments. This allows us to do inference on single CPU's, and still harness the power of state of the art models such as BERT. Traditionally, BERT in production have required cluster of GPU:s to handle high throughput applications.

Secondly, it seems that BERT can learn from noisy labels and transfer its language understanding onto new task. If we compare Figure 8 with 10, there is some vague evidence that the learnt representations are actually better than the labels themselves. We speculate this is due to the pre-training of BERT, which prevents it from overfitting on

our given labels - all while still "understanding" the training objective. Possibly this is a small step towards artificial general intelligence in natural language processing, mimicking the behaviour of a human.

A discovered limitation is the application of precomputed embeddings whenever it would be important to understand the interaction between the query document and the scientific paper. While the recommendations of similar articles in article-to-articles becomes a trivial problem with with pre-computed embeddings and achieve some really impressive results, there is still a lot of room for improvement when projecting patient notes into the same space as scientific papers (case-to-articles). While the averiage precision and ndcg may be acceptable, the standard deviation is quite high. A plausible reason for this is that some articles may be relevant to several different sets of patient notes. As a suitable analogy, "Barack Obama" could be the answer to several questions, such as "Who is a famous person born in the Hawaii?" as well as "What is the name of the former president of United States?". Similarly in medicine, the "interaction" between the patient case and the scientific paper must be accounted for in order to give a correct label on their internal relevance, whether that would be obtained with more/better data, or a different model.

6.1 Improvements

For future improvements, we have a few proposals.

- First - and probably most importantly - to mitigate the above mentioned problem of projecting cases and notes into the same vector space, we would probably benefit from using the interaction-based rankers as presented in section 2.4.4. This however introduces some computational and code complexity, which may not be feasible in production. However, there are some promising research on the concept of using knowledge distillation of BERT model still in preprint [50, 51], where a simple student model tries to learn from a much more complex teacher model. This is known to substantially reduce computation time with some loss of precision [52].

- Second, it may be interesting to investigate how the interaction between users and the accumulated knowledge during the discussion contribute go retrieval of pa-

pers. The current implementation takes no account of inter-post relations, but it is likely such relationships are of importance when recommending articles.

- Finally, relevance in recommendation system is to some degree subjective [53], while the usefulness of it is closely related to the final and overall user experience of the system [54]. Appropriately, the user experience and the usability of the system should ideally be investigated in a usability testing. Studies have shown that somewhere between 5 and 50 participants are enough to uncover all the problem associated to a solution [55].

  In this thesis, we've focused on topical similarity above other metrics (which in itself is subjective), whereas in medicine it would be safe to assume that meta data about the scientific paper also play an important part of assessing its relevance. Importantly, some types of scientific papers (such as meta and literature reviews) are considered more statistically significant than others [56]. In addition, some may only want recommendations of scientific papers specifically describing the collaboratively confirmed diagnosis, whereas some would find papers of differential diagnosis also interesting. Likewise, the year of publication, the journal, the number of citations, publication type and the author may play an important role as well.

6.2 Other use cases

In the medical domain, the setup outlined in thsis thesis can probably be used to identify similar cases to a selected one, in order to know what doctors with similar cases have concluded. Additionally, we suspect the insights gained from this thesis can transferred into other domains, including law and patent search. For instances, someone writing a patent suggestion, might be interested in a response with similar patents returned back. Similarly, someone formulating a legal case may be interested in the automatic retrieval of similar cases, for the discovery of legal precedents.

A more theoretical use case is the inverse application of precomputed embeddings in passage reranking, that we first dicussed in chapter 2.4.4. If we know X and Y are very similar to each other, we may not need a very complex model to compute the score for

both *X and Y*. Likewise, if we have discarded A, we can confidently discard B if it is very similar to A. Similar to Optimal Search Theory, we can quickly exclude parts of vector space where we are not likely to find any relevant articles, and move in the direction where we are likely to find them. Diving the search plane into several grids, we could potentially constrain the problem to scale logarithmically with the number of documents [57].

7 Conclusion

In this thesis, we have explored some of the limitation and possibilities for using the statistical language model BERT, to provide recommendations of scientific articles. Our conclusion is that precomputation of text embeddings offer a feasible alternative to using interaction-based approach. Our implementation fell short in situations where the interaction between texts from different domains, but achieved quite remarkable results on text passages from same domain. Perphaps most clearly, this thesis offers several perspectives on the digitalisation of evidence-based health care.

References

[1] I. Masic, M. Miokovic, and B. Muhamedagic, "Evidence based medicine - new approaches and challenges," *Acta Inform Med*, vol. 16, no. 4, pp. 219–225, 2008.

[2] L. P. Lavelle, R. M. Dunne, A. G. Carroll, and D. E. Malone, "Evidence-based Practice of Radiology," *Radiographics*, vol. 35, pp. 1802–1813, Oct 2015.

[3] M. Kritz, M. Gschwandtner, V. Stefanov, A. Hanbury, and M. Samwald, "Utilization and perceived problems of online medical resources and search tools among different groups of European physicians," *J. Med. Internet Res.*, vol. 15, p. e122, Jun 2013.

[4] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Computational Intelligence Magazine*, vol. 13, pp. 55–75, 08 2018.

[5] S. Ma, C. Zhang, and X. Liu, "A review of citation recommendation: from textual content to enriched context," *Scientometrics*, vol. 122, no. 3, pp. 1445–1472, 2020.

[6] K. P. Murphy, *Machine learning: a probabilistic perspective*. Cambridge, MA: MIT Press, 2012.

[7] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 39, no. 1, pp. 43 – 62, 1997.

[8] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016.

[9] Z. Zhang, "Introduction to machine learning: k-nearest neighbors," *Ann Transl Med*, vol. 4, p. 218, Jun 2016.

[10] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[11] M. Stéphane, "Chapter 10 - compression," in *A Wavelet Tour of Signal Processing (Third Edition)* (M. Stéphane, ed.), pp. 481 – 533, Boston: Academic Press, third edition ed., 2009.

[12] G. G. Chowdhury, "Natural language processing," *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 51–89, 2003.

[13] J. J. Webster and C. Kit, "Tokenization as the initial phase in nlp," in *Proceedings of the 14th Conference on Computational Linguistics - Volume 4*, COLING '92, (USA), p. 1106–1110, Association for Computational Linguistics, 1992.

[14] J. T. Goodman, "A bit of progress in language modeling," *Computer Speech Language*, vol. 15, no. 4, pp. 403–434, 2001.

[15] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, p. 1137–1155, Mar. 2003.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 3111–3119, Curran Associates, Inc., 2013.

[17] K. Cho, B. v. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation."

[18] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018.

[19] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does BERT look at? an analysis of BERT's attention," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, (Florence, Italy), pp. 276–286, Association for Computational Linguistics, Aug. 2019.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[21] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015.

[22] M. Deshpande and G. Karypis, "Item-based top- n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[23] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of Documentation*, vol. 60, no. 5, p. 503–520, 2004.

[24] H. Peng, J. Liu, and C.-Y. Lin, "News citation recommendation with implicit and explicit semantics," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Berlin, Germany), pp. 388–398, Association for Computational Linguistics, Aug. 2016.

[25] J. E. Ramos, "Using tf-idf to determine word relevance in document queries," 2003.

[26] X. Ye, H. Shen, X. Ma, R. Bunescu, and C. Liu, "From word embeddings to document similarities for improved information retrieval in software engineering," in *Proceedings of the 38th International Conference on Software Engineering*, ICSE '16, (New York, NY, USA), p. 404–415, Association for Computing Machinery, 2016.

[27] W. Huang, Z. Wu, C. Liang, P. Mitra, and C. L. Giles, "A neural probabilistic model for context based citation recommendation," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, p. 2404–2410, AAAI Press, 2015.

[28] W. Croft, *Evolution: Language Use and the Evolution of Languages*, pp. 93–120. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[29] C. Bhagavatula, S. Feldman, R. Power, and W. Ammar, "Content-based citation recommendation," *CoRR*, vol. abs/1802.08301, 2018.

[30] F. Torabi Asr, R. Zinkov, and M. Jones, "Querying word embeddings for similarity and relatedness," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 675–684, Association for Computational Linguistics, June 2018.

[31] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *CoRR*, vol. abs/1310.4546, 2013.

[32] G. R. Koch, "Siamese neural networks for one-shot image recognition," 2015.

[33] C. Jeong, S. Jang, H. Shin, E. L. Park, and S. Choi, "A context-aware citation recommendation model with BERT and graph convolutional networks," *CoRR*, vol. abs/1903.06464, 2019.

[34] R. Nogueira and K. Cho, "Passage re-ranking with BERT," *CoRR*, vol. abs/1901.04085, 2019.

[35] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, "Understanding the behaviors of BERT in ranking," *CoRR*, vol. abs/1904.07531, 2019.

[36] Y. Ren, S. Yoo, and A. Hoisie, "Performance analysis of deep learning workloads on leading-edge systems," *CoRR*, vol. abs/1905.08764, 2019.

[37] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019.

[38] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *CoRR*, vol. abs/1503.03832, 2015.

[39] M. S. Handcock, A. E. Raftery, and J. M. Tantrum, "Model-based clustering for social networks," *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, vol. 170, no. 2, pp. 301–354, 2007.

[40] W. Lu, W. Lu, J. Janssen, J. Janssen, E. Milios, E. Milios, N. Japkowicz, N. Japkowicz, Y. Zhang, and Y. Zhang, "Node similarity in the citation graph," *Knowledge and Information Systems*, vol. 11, no. 1, pp. 105–129, 2007;2006;.

[41] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, (San Francisco, California, USA), pp. 855–864, ACM Press, 2016.

[42] C. Xiao, J. Ye, R. M. Esteves, and C. Rong, "Using spearman's correlation coefficients for exploratory data analysis on big dataset: Using spearman's correlation coefficients for exploratory data analysis," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 14, pp. 3866–3878, 2016.

[43] H. Dalianis, *Characteristics of Patient Records and Clinical Corpora*, pp. 21–34. Cham: Springer International Publishing, 2018.

[44] H. A. Mohamed Hassan, G. Sansonetti, F. Gasparetti, A. Micarelli, and J. Beel, "Bert, elmo, use and infersent sentence encoders: The panacea for research-paper recommendation?," 09 2019.

[45] H. Jia and E. Saule, "An analysis of citation recommender systems: Beyond the obvious," pp. 216–223, ACM, 2017.

[46] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *arXiv preprint arXiv:1702.08734*, 2017.

[47] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, 09 2019.

[48] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[49] A. Coenen, E. Reif, A. Yuan, B. Kim, A. Pearce, F. B. Viégas, and M. Wattenberg, "Visualizing and measuring the geometry of BERT," *CoRR*, vol. abs/1906.02715, 2019.

[50] O. Barkan, N. Razin, I. Malkiel, O. Katz, A. Caciularu, and N. Koenigstein, "Scalable attentive sentence-pair modeling via distilled sentence embedding," 2019.

[51] W. Lu, J. Jiao, and R. Zhang, "Twinbert: Distilling knowledge to twin-structured bert models for efficient retrieval," 2020.

[52] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[53] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, p. 5–53, Jan. 2004.

[54] J. A. Konstan and J. Riedl, "Recommender systems: From algorithms to user experience," *User Modeling and User-Adapted Interaction*, vol. 22, pp. 101–123, apr 2012.

[55] J. M. C. Bastien, "Usability testing: a review of some methodological and technical aspects of the method," *International Journal of Medical Informatics*, vol. 79, no. 4, pp. e18–e23, 2008;2010;.

[56] L. P. Lavelle, R. M. Dunne, A. G. Carroll, and D. E. Malone, "Evidence-based practice of radiology," *RadioGraphics*, vol. 35, pp. 1802–1813, Oct. 2015.

[57] D. Kirkpatrick, "Optimal search in planar subdivisions," *SIAM Journal on Computing*, vol. 12, no. 1, pp. 28–35, 1983.

A Appendix

Case Meta Data:

Title: saddle embolism

Section: Chest Imaging

Educational Value: 4 (1 vote)

DICOM tags: #50y #female #CT

Original Post:

User: ??????

DateTime: 2019-09-09 10:48:33

Tags: #pulmonary_embolus

Description: patient on intensive care unit, ETT, ventilated acute deterioration CT chest : spontaneaous hyperdense pulmonary arteries Patient collapsed during the CT acquisition // reanimation started at CT table #pulmonary_embolus #saddle_embolism no contrast administered, actually reanimation started on CT table

Suggested Diagnosis: #pulmonary_embolus #saddle_embolism

Confidence: 5

Agree: 4

Replies

Reply 1:

Tags: #pulmonary #embolism

DateTime: 2019-09-12 10:48:33

User: ??????

Comment:

Interesting case, Specially interesting to note the low sensitivity and high

specificity for using this sign.

Previously reported in *The validity of hyperdense lumen sign in non-contrast chest CT scans in the detection of pulmonary thromboembolism*. Tatco VR, Piedad HH. Int J Cardiovasc Imaging. 2011 Mar;27(3):433-40. https://www.ncbi.nlm.nih.gov/pubmed/20658266?report=abstract

Suggested Diagnosis: #pulmonary #embolism

Confidence: 4

Agree: 1

Reply 2:

DateTime: 2019-09-13 10:48:33

User: ??????

Comment:

Very nice case, thnx for sharing.

Reply 3:

DateTime: 2019-09-14 10:48:33

User: ??????

Comment:

@?????? It is a very nice case. What is the mean HU on the other side? It would be good to know that to make the case more obvious. Did you try to do the image reconstruction in thin slices? In neuroradiological imaging, it is well known that thrombi are more visual when the CCTs are reconstructed in thin slices.

Reply 4:

DateTime: 2019-09-15 10:48:33

User: j????

Comment:

@?????? thicher slices here just to try to show the finding in a "still" image diagnosis was made in thin slices while pt was being resuscitated on table middle-of-the-night diagnosis

Reply 5:

DateTime: 2019-09-16 10:48:33

User: saif-afat

Comment:

@?????? Thank you for sharing this information with us. It would be amazing if you could upload the thin slices to this case. Best regards, ?????