KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Computer Science with specialization in Machine Learning

Second cycle, 30 credits

# Medical image captioning based on Deep Architectures

## Medicinsk bild textning baserad på Djupa arkitekturer

**Georgios Moschovis**

Stockholm, Sweden, 2022

## Author

Georgios M. Moschovis <geomos@kth.se>
Machine Learning, Electrical Engineering and Computer Science
KTH Royal Institute of Technology

## Medical image captioning based on Deep Architectures

### Medicinsk bild textning baserad på Djupa arkitekturer

Neural Dynamics Lab, School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden

## Examiner

Olov Engwall <engwall@kth.se>
Department of Intelligent Systems, School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

## Supervisor

Erik Fransén <erikf@kth.se>
Science for Life (SciLife) Lab, School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

# Abstract

Diagnostic Captioning is described as "the automatic generation of a diagnostic text from a set of medical images of a patient collected during an examination" [59] and it can assist inexperienced doctors and radiologists to reduce clinical errors or help experienced professionals increase their productivity. In this context, tools that would help medical doctors produce higher quality reports in less time could be of high interest for medical imaging departments, as well as significantly impact deep learning research within the biomedical domain, which makes it particularly interesting for people involved in industry and researchers all along.

In this work, we attempted to develop Diagnostic Captioning systems, based on novel Deep Learning approaches, to investigate to what extent Neural Networks are capable of performing medical image tagging, as well as automatically generating a diagnostic text from a set of medical images. Towards this objective, the first step is concept detection, which boils down to predicting the relevant tags for X-RAY images, whereas the ultimate goal is caption generation.

To this end, we further participated in ImageCLEFmedical 2022 evaluation campaign, addressing both the concept detection and the caption prediction tasks by developing baselines based on Deep Neural Networks; including image encoders, classifiers and text generators; in order to get a quantitative measure of my proposed architectures' performance [28]. My contribution to the evaluation campaign, as part of this work and on behalf of *NeuralDynamicsLab*[1] group at KTH Royal Institute of Technology, within the school of Electrical Engineering and Computer Science, ranked 4th in the former and 5th in the latter task [55, 68] among 12 groups included within the top-10 best performing submissions in both tasks.

## Keywords

Artificial Neural Networks, Deep Learning, Speech and language technology, Natural Language Processing (NLP), Deep networks, Generative deep networks, Convolutional neural networks (CNN), Text generation, Information retrieval, Diagnostic captioning, Image captioning, concept prediction, classification, image encoders, transformers, Encoder-Decoder architecture, abstractive summarization.

---

[1] https://www.csc.kth.se/∼erikf/

# Abstrakt

Diagnostisk textning avser automatisk generering från en diagnostisk text från en uppsättning medicinska bilder av en patient som samlats in under en undersökning och den kan hjälpa oerfarna läkare och radiologer, minska kliniska fel eller hjälpa erfarna yrkesmän att producera diagnostiska rapporter snabbare [59]. Därför kan verktyg som skulle hjälpa läkare och radiologer att producera rapporter av högre kvalitet på kortare tid vara av stort intresse för medicinska bildbehandlingsavdelningar, såväl som leda till inverkan på forskning om djupinlärning, vilket gör den domänen särskilt intressant för personer som är involverade i den biomedicinska industrin och djupinlärningsforskare.

I detta arbete var mitt huvudmål att utveckla system för diagnostisk textning, med hjälp av nya tillvägagångssätt som används inom djupinlärning, för att undersöka i vilken utsträckning automatisk generering av en diagnostisk text från en uppsättning medi-cinska bilder är möjlig. Mot detta mål är det första steget konceptdetektering som går ut på att förutsäga relevanta taggar för röntgenbilder, medan slutmålet är bildtextgenerering.

Jag deltog i ImageCLEF Medical 2022-utvärderingskampanjen, där jag deltog med att ta itu med både konceptdetektering och bildtextförutsägelse för att få ett kvantitativt mått på prestandan för mina föreslagna arkitekturer [28]. Mitt bidrag, där jag representerade forskargruppen *NeuralDynamicsLab*[2], där jag arbetade som ledande forskningsingenjör, placerade sig på 4:e plats i den förra och 5:e i den senare uppgiften [55, 68] bland 12 grupper som ingår bland de 10 bästa bidragen i båda uppgifterna.

## Nyckelord

Neurala nätverk, Djup inlärning, Tal-och språkteknologi, naturlig språkbehandling, djup neurala nätverk, generativa djupa nätverk, konvolutionella neurala nätverk, Textgenerering, Informationssökning, Diagnostisk textning, Bildtextning, konceptförutsägelse, klassificering, bildkodare, transformatorer, kodaravkodararkitektur, abstrakt sammanfattning.

---

[2] https://www.csc.kth.se/∼erikf/

# Acknowledgements

First and foremost, I would like to express all my gratitude to my parents for both their practical help and mental support, since they not only provided me with all the necessary hardware and converted an old storeroom into a fully-equipped study room but also stood by me every time I felt demotivated. Without their mindful thinking and care, this thesis would not have been completed. Moreover, I would like to thank my friends for their continuous support and motivation, my psychiatrist Dr. Nikolaos Zouros for helping me deal with mild autism spectrum disorder (Asperger syndrome as per ICD 10) and remain productive, professor Ioannis Pavlopoulos and Vasiliki Kougia for providing useful references and practical tips as well as my examiner Olov Engwall for offering useful advice and guidance. Last but not least, I am extremely thankful to my supervisor Erik Fransén, whom I felt as a parent while being in Stockholm since apart from the scientific instructions, he was extremely helpful with all bureaucratic procedures, always ensured that I had access to study places within school premises and without exaggeration cared about every aspect that could affect the conduction of this master thesis and its defence.

Steven Hawking said ”intelligence is the ability to adapt to change” in his graduation speech at Oxford University, a claim that heavily influenced me to reconsider my way of thinking, get out of my comfort zone and start overcoming challenges from a different perspective. Completing this work is the informal conclusion of a long process of stress but also change, adaptation, realization of my inner strengths and understanding of the world. It is the end of a process undoubtedly making me a more confident professional and a better person. Taking home profound knowledge and social skills acquired during the process, the determination to leave behing stereotypes built to resist to the pandemic lockdowns and the experience of academic excellence, I hope this work will be useful to other researchers.

# Contents

1 Introduction 1

1.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
1.2 Problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
1.3 Purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.4 Benefits, Ethics and Sustainability . . . . . . . . . . . . . . . . . 3
1.5 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

2 Theoretical Background 5

2.1 Neural Network Generative Architectures . . . . . . . . . . . . . . 5
  2.1.1 Recurrent Neural Networks . . . . . . . . . . . . . . . . . . . 5
  2.1.2 Transformer Networks . . . . . . . . . . . . . . . . . . . . . . 6
2.2 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  2.2.1 Datasets . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

3 Research Methodology 11

3.1 Data Processing . . . . . . . . . . . . . . . . . . . . . . . . . . 11
  3.1.1 ImageCLEFmedical 2022 data split and statistics . . . . . . . . 11
  3.1.2 Other publicly available datasets . . . . . . . . . . . . . . . . 12
3.2 Training Regimes . . . . . . . . . . . . . . . . . . . . . . . . . . 13
  3.2.1 Adaptive Moments optimizer . . . . . . . . . . . . . . . . . . . 13
  3.2.2 Adaptive Moments with weight decay optimizer . . . . . . . . . 14
  3.2.3 Glorot initialization . . . . . . . . . . . . . . . . . . . . . . 14
3.3 Retrieval Augmented Generation . . . . . . . . . . . . . . . . . . 14
  3.3.1 Dense Passage Retriever . . . . . . . . . . . . . . . . . . . . . 16
  3.3.2 Generative Transformers and Denoising Autoencoders . . . . . 16
  3.3.3 RAG token . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
3.4 Evaluation goals and delimitations . . . . . . . . . . . . . . . . . 20

3.5 Significance tests for experiments outcomes . . . . . . . . . . . . . . . . 21

4 Model selection and hyperparameter tuning 22

4.1 ImageCLEFmedical 2022 contributions . . . . . . . . . . . . . . . . . . . 22

4.1.1 Concept prediction subtask . . . . . . . . . . . . . . . . . . . . . . 23

4.1.2 Caption generation subtask . . . . . . . . . . . . . . . . . . . . . . 30

4.1.3 Concept Detection Performance summary . . . . . . . . . . . . 33

4.1.4 Caption Generation Performance summary . . . . . . . . . . . . 33

4.2 Additional experiments on IU X-RAY . . . . . . . . . . . . . . . . . . . . 35

4.2.1 Class imbalance on IU X-RAY: detection and handling . . . . . 36

4.2.2 Extended frequency analysis of IU X-RAY captions . . . . . . . 38

4.3 Statistical Significance Analysis outcomes for our ImageCLEFmedical 2022 contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

4.3.1 Significance tests for all experiments outcomes . . . . . . . . . 41

4.3.2 Significance tests for submitted baselines . . . . . . . . . . . . . 42

4.4 Statistical Significance Analysis outcomes for our additional experiments on IU X-RAY . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

4.4.1 Baselines comparison and performance ordering . . . . . . . . 44

4.4.2 Statistical significance tests to verify class imbalance . . . . . . 45

5 Discussion and conclusions 46

5.1 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

5.2 Best submissions in ImageCLEF medical 2022 . . . . . . . . . . . . . . 48

5.3 Indicative generations of our systems . . . . . . . . . . . . . . . . . . . 49

References 53

6 Experimental results and details 66

6.1 ImageCLEFmedical 2022 contributions . . . . . . . . . . . . . . . . . . . 66

6.1.1 Concept Detection Performance summary . . . . . . . . . . . . 66

6.1.2 Caption Generation Performance summary . . . . . . . . . . . . 67

6.2 Additional experiments on IU X-RAY . . . . . . . . . . . . . . . . . . . . 70

6.2.1 Diagnostic Captioning experiments . . . . . . . . . . . . . . . . 70

7 Additional results and statistics 73

# Chapter 1

# Introduction

## 1.1 Background

One of the most exciting technological aspects nowadays is the impressive potential of Machine Learning in transforming the world we live in, primarily because of its exciting resurgence through Deep Learning (DL). The increasing size of biomedical data has allowed researchers demonstrate its evolving capabilities in biomedical applications, through the development of advanced computing and imaging systems in biomedical engineering, machine learning-based biomedical data mining algorithms [41] but also baselines for Diagnostic Captioning. The latter task has recently attracted researchers' attention towards the goal of reducing the time required by a doctor or radiologist in order to produce medical texts and the amount of clinical errors, but also increasing the throughput of medical imaging departments [59].

## 1.2 Problem

Medical doctors are prone to error due to their increased workload and high pressure, because of increasing demands on doctors to do documentation and continuous cost-saving pressure in health care sectors. These factors consequently affect the quality of their produced medical reports. The core idea addressed in this work is to develop a Diagnostic Captioning model, based on novel Deep Learning architectures, in order to help doctors and radiologists produce more accurate medical diagnoses and gain some improvements in terms of both accuracy and speed, by providing them tools that are capable of automatically generating a draft diagnostic text from a set of medical images. While these tools are being further trained and acquire knowledge, by incorporating the concept of active learning [65], the aforementioned improvements should become even more significant.

Accuracy is extremely crucial when filling medical reports; not omitting any important findings and not referring to wrong findings; due to lack of concentration, time, spur or experience. Accuracy may also be addressed to as lack of medical errors. Speed is another important factor to take into consideration when producing a medical report; a draft version of the diagnosis, automatically written by a sufficiently trained system, would significantly reduce the time required by a doctor or radiologist but also increase the throughput of medical imaging departments. In this context, follows a high-level illustration of the process in order to complete a medical report, involving the medical doctor and the expert system.

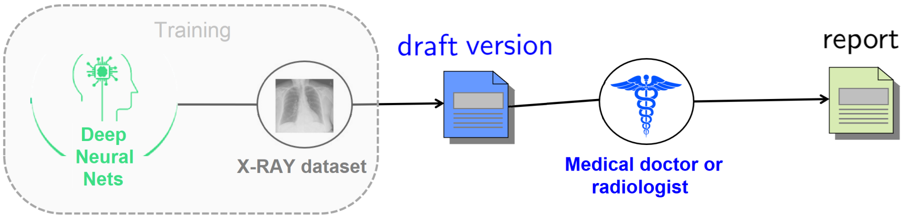


Figure 1.2.1: Visualization of the Diagnostic Captioning pipeline involving both the medical doctor and the Deep Learning system, which we train using Neural Networks. A draft version of the report is provided to the medical doctor to complete, who also considers the patient's history.

## 1.3 Purpose

In this work, the core aim was to develop Diagnostic Captioning (DC) systems, based on novel Deep Learning approaches, to investigate to what extent Neural Networks are capable of automatically generating a diagnostic text from a set of medical images and how much their interpretation of medical images can assist doctors and radiologists to increase the quality and speed of producing medical diagnoses and meanwhile reduce the time needed. This is also associated to an increased throughput of medical imaging departments [59]. Towards this objective, the first step is concept prediction that boils down to predicting the illnesses associated to X-RAY images, while the ultimate goal is caption generation.

Towards the concept prediction subtask, we employed Convolutional Neural Network (CNN) image encoders to codify the images into dense representations, precisely either pre-trained on ImageNet [70] then finetuned in the X-RAY images or directly trained in the respective dataset, according to the pipelines described in chapter 4. These CNN models are also extensively referred to as "backbone networks", which is explained in section 1.5 as well. Ensembles of CNN encoders shall also be tested to seek for diversity and in order to exploit the "Wisdom of the crowd" [76] for the fine-tuned models that we propose for this subtask.

For the caption generation subtask, my proposed baseline models are both based on the transformer architecture [81] and rely on either abstractive summarization, through a model called Pegasus [91] or the pipeline proposed as Retrieval Augmented Generation (RAG) [45]. Precisely, in the latter approach, the idea was to combine the massive success of sequence-to-sequence models in NLP with the strengths of Dense Passage Retriever (DPR) [34], which characterizes modern Information Retrieval by using a FAISS index [32] for further efficiency.

## 1.4 Benefits, Ethics and Sustainability

Development of Diagnostic Captioning systems based on novel DL architectures could have both positive and negative societal impacts. My proposed work, for example, may be used for analyzing medical image data in undeveloped regions or countries under development. This is closely related to the $3^{rd}$ goal of United Nations Sustainability Goals about ensuring good health and well-being and the $10^{th}$ goal about reduced inequalities. On the other hand, privacy issues might arise from the use of medical data and "concerns over the sensitive information security and privacy" [1]. Those may also be related to the General Data Protection Regulation (GDPR) and the EU legislation on data privacy and protection (679/2016, 680/2016, 2018/1725).

Furthermore, we attempted to develop Diagnostic Captioning baselines based on deep architectures in order to question how much their interpretation of medical images can assist doctors and radiologists to produce better quality diagnoses but also at an increased throughput [59]. A system that can assist in saving doctor time, possibly also by incorporating the concept of active learning [65], would eventually contribute to a more sustainable society despite the fact that training a deep network indeed leads to an energy cost until incorporating parametric knowledge.

## 1.5 Methodology

One of the principal components in the proposed architectures that is shared for both subtasks includes the image encoders. They constitute state-of-the-art architectures, pretrained on ImageNet classification dataset [70], which have been obtained through `torchvision` models library to perform inference. Then, any additional components such as a multi-label classification head or a caption generation architecture have been appended to the output of the image encoder; in this content these models are referred to as “backbone networks”.

Some Convolutional Neural Network encoders that have been attempted to use include variants of AlexNet [40], ResNet [27], DenseNet [25], VGG [73] and EfficientNet [78], which have been obtained from `torchvision` models library as mentioned above. We also experimented with Vision Transformers (ViT) [17], thus another architectural choice, the performance obtained was poor however compared to CNN encoders. That outcome is in line with the observation in [4] that Vision Transformers and “Hybrid-ViT architectures are inferior to the CNN-based ones”. The above summarize the first step in the design of image encoders that is model selection based on their performance on a separate development set.

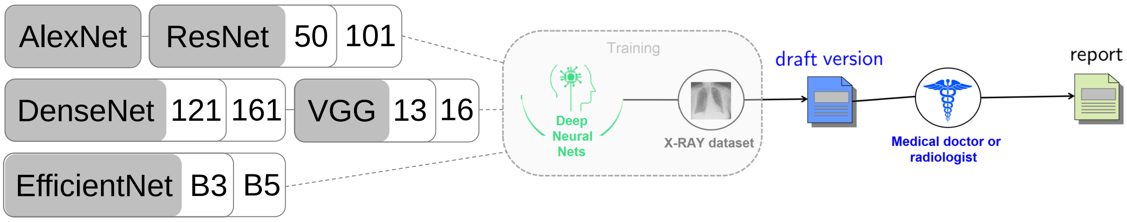


Figure 1.5.1: Detailed visualization of the Diagnostic Captioning pipeline.

Moreover, model selection shall be followed by a model collaboration design principle, based on ensemble learning. In this case, we have therefore used the aforementioned models as *members of the ensemble* or *weak learners* in a pool of encoders trained with different parameter values (such as learning rates, decision thresholds for the positive class, number of epochs), as well as based on different architectures, to seek for diversity and exploit the “Wisdom of the crowd” [76] for the fine-tuned models. In this context, we take into consideration the “votes” of all the different CNNs by averaging their outputs to make guesses on the assigned tags or make decisions on the generated captions’ tokens.

# Chapter 2

# Theoretical Background

## 2.1 Neural Network Generative Architectures

### 2.1.1 Recurrent Neural Networks

The neural networks that deal with sequences are Recurrent Neural Networks (RNNs). They are distinguished from Feed-Forward Networks by that feedback loop connected to their past decisions. RNNs take as their input, not just the current input sample, but also the information they have perceived previously and have two sources of input, the present and the recent past, which are combined to determine the output. Hence, it is often claimed that Recurrent Neural Networks have memory [11] as they are able to store information.

However, although RNNs are capable to learn how to use past information if the gap between the relevant information and the place that it is needed is small; there might be cases we need more content. It is entirely possible for the gap between the relevant information and the point where it is needed to become also very large. Unfortunately, as the gap grows, RNNs become unable to learn to connect all the information [11]. Adding LSTM or GRU cells yields models that are typically better than the so-called ”vanilla” RNNs at remembering long-term information [26] but there is an upper limit in their performance as well.

The technique to train RNNs is called “backpropagation through time” or BPTT, and it consists of a generalization of back-propagation for feed-forward networks that is based on their history. This process however is prone to vanishing gradients problem.

In a nutshell, the problem comes from the fact that at each time step during training we are using the same weights to calculate the output of the RNN, thus it experiences difficulty in memorizing elements from far away in the sequence and makes predictions based on the most recent ones.

Both long short-term memory (LSTM) [26] and Gated Recurrent Unit (GRU) [10] cells, constitute complex RNN cells that tackle the vanishing gradients problem. Each cell consists of a *Forget Gate*, an *Input Gate* and an *Output Gate*. The *input gate* controls the extent to which a new value flows into the cell. The *forget gate* controls the extent to which a value remains in the cell. Finally, the *output gate* controls the extent to which the value in the cell is used to compute the output activation of the LSTM or GRU unit. LSTM cells are similar to GRU cells but they contain more complex computations inside their gates.

### 2.1.2 Transformer Networks

To improve performance of modern NLP systems, the transformer architecture that revolutionized the field was proposed in 2017 [81] and is based solely on attention mechanisms, dispensing with recurrence and convolutions entirely and weighing the influence of different parts of the input data. Attention is a concept that improved the performance of all downstream NLP tasks –an ubiquitous method in modern deep learning models [2] that matches a query and a set of key-value pairs to an output; all of them vectorized. If we denote the queries as row vectors of a matrix $Q$, as well as the keys as part of a matrix $K$ and the values contained in a matrix $V$, then attention is computed according to the following formulas:

$$\mathcal{A}(Q,K) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)$$

$$\text{Attention}(Q,K,V) = \mathcal{A}(Q,K)V = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V$$

As most competitive neural sequence transduction models, transformer networks have an encoder-decoder structure [5]. Thus, that means given an input sequence of words such as $\mathcal{W} = \{w_1, w_2, ..., w_N\}$, the encoder maps it to a sequence of continuous representations $\mathcal{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_N\}$ for the decoder to produce an output sequence $\hat{\mathcal{W}} = \{\hat{w}_1, \hat{w}_2, ..., \hat{w}_N\}$ in an auto-regressive manner [23], consuming the previously generated words that we denote by $\hat{w} \in \hat{\mathcal{W}}$ [69]. They address the problem of previous network architectures poorly retaining contextual relationships across long texts and can be highly parallelized, train models with billions of parameters at a higher rate, and use contextual clues to reduce ambiguity issues.

### Transformers for abstractive summarization

One of the most important steps in neural or automatic text summarization is sentence extraction, which boils down to generating a summary by identifying and subsequently concatenating the most salient text units in a document while maintaining key concepts and information [9]. Precisely, abstractive summarization relates to text generation that summarizes the original content while it captures "key ideas and elements of the source text, usually involving significant changes and paraphrases of text from the original source sentences" without changing their meaning [62]. Due to the difficulty of information extraction and automatic text generation, abstractive summarization has been considered a rather complex problem [33] but transformer networks [81] have achieved inspiring results. In our work, we use a model called Pegasus [91] that is based on the transformer architecture, reporting surprising performance especially on low-resource summarization.

## 2.2 Related Work

During the last decades, extensive research has been conducted regarding Diagnostic Captioning. Back in 2004, structured support vector machines have been attempted to use for generating semantic tags for regions in an organ or tissue that have suffered damage through injury or disease [79], while in 2012 an ontology-based approach was followed towards language generation from cardiological findings detected in X-RAY images [80]. It was not until 2015, when a CNN backbone network was first used to classify 3-dimensional pixels into different categories given as input diagnostic tags extracted from medical reports [71]. One year later, another CNN architecture was proposed for detecting suspicious pixels regions in the X-RAY images, followed by a classification head consisting of fully-connected layers to assign abnormality labels to those regions [38].

Ever since, CNN backbone networks have been impressively preferred in Diagnostic Captioning research, which is in line with their performance in classification [51] and generic image captioning [77], as well as in digital holography [12], possibly combined with Recurrent Neural Network (RNN) architectures; trained end-to-end. The latter approach was first used for keyword generation, featuring LeNet [42] as image encoder and attempts with both Long-Short Term Memory (LSTM) [26] and Gated Recurrent Unit (GRU) cells [10] for the text generation. Precisely, the LeNet image encoder was used for the prediction of relevant concepts among 17 tags during pre-training, which were extended to 57 subcategories according to clusters created by $k$-means [53] during fine-tuning. For text generation, Recurrent Neural Networks with LSTM or GRU cells were used to address the vanishing gradients problem described in section 2.1.1 for the vanilla RNNs, however the proposed architecture only managed to produce keywords for the X-RAY images' contents [72].

The aforementioned architecture has been attempted to use for caption generation as well, however a ResNet-based encoder [27] was employed to codify the X-RAY images into dense representations and only LSTM cells were used in the generator to produce the corresponding captions. Although feature extraction was proved possible from the LSTM cells' history, this setting performed poorly in CHEST X-RAY 14 dataset [86] in terms of evaluation with BLEU scores [57] when attempting to produce well reasoned medical reports instead of just keywords [87].

Furthermore, to overcome the issues attributed to RNN training, stacked LSTMs were replaced by a hierarchical LSTM [19] accompanied by a VGG-19 backbone network [73] with a Feed-Forward Network in the image encoder output, to predict probabilities for the different tokens. Those with the highest probabilities were fed to the hierarchical LSTM decoder, where the lower level cells produced dense vector representations of the encoded topics and attention was also used in their states to contextualize visual features and perform selection of areas within the X-RAY image and visual features, resulting in sentence embeddings. When the lower level cells received `END` signals, the higher level cells started generating the medical report per token until their `END` signals were produced [30]. This work has borrowed ideas from relevant models addressing both diagnostic captioning [94, 95] and generic image captioning [16, 84, 89, 90], while it outperforms similar architectures targeting the latter, even after they are fine-tuned in X-RAY images for caption generation.

Other works have attempted to make use of attention for both concept detection and caption generation tasks. Considering the former, TandemNet [94] was proposed that was later generalized to MDNet [95] that is capable of addressing the latter task as well.

These models include ResNet-based encoders [27] and LSTM decoders with attention for concept detection and text generation for a few different symptoms. More recently, modern CNNs such as DenseNet121 [25] and ResNet-152 [27] have been preferred to use as backbone networks to output concepts for chest X-RAY images [64] or predict pathological abnormalities and detect their locations [85] over other networks such as LeNet [42] or shift-invariant CNNs [92].

Moreover, some works take advantage of the class-imbalance issues that occur in some datasets for DC, such as the Indiana University chest X-RAY Collection [14] (IU chest X-RAY dataset) that is further described in section 3.1.2 and classify the captions as positive and negative findings. Using the Frontal Pelvic X-RAYs, including 50,363 images with simplified medical reports, one approach using a DenseNet encoder [25] and a stacked two-layered LSTM with attention, produced always the same caption for negative findings and only five keywords for positive findings, while it performed extremely well in terms of BLEU scores [57]. The employed approach was Template-based generation, which requires the model to fill-in a document template, precisely predict five tags in the case of positive findings [20].

In addition, there have been several attempts to combine Deep Networks training with other learning approaches, such as Reinforcement Learning (RL). The REINFORCE algorithm [88] used for optimal control, which is based on Robbins-Monro Stochastic Approximation method [67], has been attempted to use both in the context of generic image captioning [66] and Diagnostic Captioning. In the latter case, RL is involved to decide whether a frequent caption will be provided or the diagnostic text will be generated by an Encoder-Decoder architecture that consists of a DenseNet121 image encoder [25] and a hierarchical LSTM [19] text generator, according to either a reward function based on CiDER score [46, 82] or extracted by comparing system with human-authored tags, for instance using CheXpert [29, 50].

Last but not least, Information Retrieval based methods have demonstrated promising potential in Diagnostic Captioning and Language Modelling in general [35, 36]. The most simplified example is the 1-NN baseline [50], which has also been included in our work and is extensively described in sections 4.1.1 and 4.1.2. In summary, this simple model assigns the tags of the visually most similar image from the training set as the output for concept prediction and produces the diagnostic text of the visually most similar image from the training set as the output for caption generation [59]. A more novel approach, based on retrieval that uses transformer networks [81], “decomposes medical report generation into explicit medical abnormality graph learning and subsequent natural language modeling” and by doing so outperforms other captioning approaches [47]. Our proposed model, employing Retrieval Augmented Generation (RAG) [45] has attempted to combine the success of parametric sequence-to-sequence models with the strengths of Dense Passage Retriever [34], which designates modern Information Retrieval [45].

### 2.2.1 Datasets

One of the main reasons why Diagnostic Captioning has recently attracted researchers’ attention is the increasing amount of biomedical datasets, including PEIR GROSS [30], the data used in ImageCLEF Medical evaluation campaign of previous years [18, 21], the extended Radiology Objects in COntext (ROCO) dataset, a subset of which was used for ImageCLEF Medical 2022 evaluation campaign [28, 60], Indiana University chest X-ray (IU X-RAY) Collection [14] and MIMIC-CXR [31], while additionally there are other datasets, i.e. provided in Bio-ASQ challenge on large-scale biomedical semantic indexing and question answering, as well as those used in related work and therefore mentioned in section 2.2.

The next chapter provides all the details about the datasets used for our experiments. In section 3.1.1, we describe the data provided in ImageCLEF Medical 2022 evaluation campaign used for our training. Precisely, we give further details about the ImageCLEF Medical 2022 concept detection and caption prediction dataset, that was provided as input to our generic image encoders or backbone networks that rely on Convolutional Neural Networks (CNN) architectures. These are popular for vision tasks on generic images, such as classification and semantic segmentation, while they are shared within all baselines, in both tasks. In section 3.1.2 we also refer to a publicly available dataset, the Indiana University chest X-RAY Collection [14] (IU X-RAY), which was used to perform additional experiments.

# Chapter 3

# Research Methodology

## 3.1 Data Processing

In this section, we describe the data provided in ImageCLEFmedical 2022. In section 3.1.1, we describe the data provided in ImageCLEF Medical 2022 evaluation campaign used for our training. Precisely, we provide details about the ImageCLEFmedical 2022 concept detection and caption prediction datasets that include images from different radiological image modalities but without including imaging modality information. What is more, in section 3.1.2 we explain our pipelines for IU X-RAY dataset [14] that was used to perform additional experiments.

### 3.1.1 ImageCLEFmedical 2022 data split and statistics

The data provided for both subtasks of ImageCLEF Medical 2022 evaluation campaign this year [28] consist of $90920$ images that constitute a subset of the extended Radiology Objects in COntext (ROCO) dataset [60], without imaging modality information. As in previous years [18, 21], the dataset originates from biomedical articles of the PMC OpenAccess subset. After merging the initially provided train and validation data, we shuffle them after manually setting the seeds to eliminate randomness in consecutive runs while tuning our hyperparameters and then keep $80\%$ as our training set, $10\%$ as our validation set to perform hyperparameter tuning and the remaining $10\%$ as our development set to perform model selection.

Since the dataset is large we perform neither cross-validation nor data-augmentation. We experimented with adding noise to the images, in the form of random rotations and translations, which however did not provide any additional benefit in our models' quantitative evaluation. Other types of noise, such as Gaussian or salt-and-pepper noise were not attempted to use as although they could have improved quantitative performance in terms of F1 or BLEU scores [57], they could also reduce visibility of the regions of interest in the X-RAY image, e.g. locations depicting an organ or tissue that suffered damage through injury or disease, limiting our proposed approaches' practical usefulness and interpretability [54].

Regarding the concept detection subtask, we detected $8374$ tags of concepts that are assigned to the X-RAY images, while each X-RAY image drawn from any of the training, validation or development set is assigned $5$ tags on average. The concepts distribution is skewed however since $4716$ concepts have extremely few occurrences. Regarding the caption prediction subtask, the total number of captions in the resulting training set is $72736$, the total number of unique captions is $70879$ and the average caption length $108$ words, including $28$ unique words. In the validation set the total number of captions is $9092$, the total number of unique captions is $8984$, the average caption length is $107$ words, including $26$ unique words. In the development set the total number of captions is $9092$, the total number of unique captions is $8977$ and the average caption length is $108$ words, including $28$ unique words.

"The concepts were generated using a reduced subset of the Unified Medical Language System (UMLS) 2020 AB release, which includes the sections (restriction levels) 0, 1, 2, and 9" [28]. The UMLS is a set of files and software that collects multiple health and biomedical vocabularies and standards to enable interoperability between computer systems. To improve the feasibility of recognizing concepts from the images, concepts were filtered based on their semantic type and concepts with very low frequency were removed. In each caption, tokens containing numbers and all punctuation were removed, captions were converted to lower-case and lemmatization was applied using spaCy toolkit [68].

### 3.1.2 Other publicly available datasets

Apart from ImageCLEF data, IU X-RAY [14] and MIMIC-CXR [31] constitute known biomedical Datasets that are publicly available and both contain medical images and diagnostic reports. Moreover, it is important to mention that they contain fields that are almost identical, therefore using them both is facilitated in this sense. The most important fields are described in section 4.2 and among those we took advantage of both IMPRESSION and FINDINGS fields where they exist; to generate the images' captions. Then, the professional shall consider the remaining fields that mostly refer to previous illnesses of the patient, to complete the draft report. Furthermore, MIMIC-CXR was not used for performing experiments in this work, however using them both should be trivial since they contain fields that are almost identical, provided that access may be granted to the latter beforehand.

One problem identified, is that very often the diagnostic reports are very similar across patients, as well the class imbalance between reports with no findings compared to other reports referring to abnormalities, but also missing information in sections which require access to unavailable previous examinations for a particular patient. A large class imbalance "may lead to misleadingly high Accuracy, which is why Precision, Recall, and $\mathrm{F}_\alpha$ scores" –mostly F1 score– are used instead in such cases, being modelled by the metrics described in section 3.4, based on traditional retrieval schemes used to model word overlap in the targets [59].

## 3.2 Training Regimes

### 3.2.1 Adaptive Moments optimizer

When performing stochastic or minibatch Gradient Descent, and the loss changes quickly at one direction and slowly at another, Gradient Descent will progress slowly along the shallow dimension and jitter along the steep one. To overcome this issue, we used Adaptive Moments (Adam) optimizer [37], so that progress along steep directions is damped and meanwhile progress along flat directions is accelerated. Adam uses exponentially decaying average to discard history but also momentum as an estimate of the first-order gradient. It includes bias corrections for first-order and second-order moments and converges rapidly after finding a local convex bowl. If $t$ represents the current time step, Adam updates are equal to:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \epsilon\frac{\mathbf{v}^{(t)}}{\delta + \sqrt{\mathbf{r}^{(t)}}}, \delta, \epsilon \in \mathbb{R}^+$$

$$\mathbf{v}^{(t+1)} = \rho_1\mathbf{v}^{(t)} + (1-\rho_1)\mathbf{g}^{(t)}, \rho_1 \in \mathbb{R}^+$$

$$\mathbf{r}^{(t+1)} = \rho_2\mathbf{r}^{(t)} + (1-\rho_2)\left(\mathbf{g}^{(t)}\right)^2, \rho_2 \in \mathbb{R}^+$$

### 3.2.2 Adaptive Moments with weight decay optimizer

Adaptive Moments with weight decay optimizer (AdamW), constitutes an improved version of Adam optimizer [52], where weight decay is performed only after controlling the parameter-wise step size and thus yields models that are capable of generalizing better. Compared to Adam optimizer, as well as other adaptive gradient algorithms, where the potential benefit of weight decay regularization is limited because "the weights do not decay multiplicatively but by an additive constant factor" [52], AdamW optimizer may overcome this issue while also training much faster than stochastic or minibatch Gradient Descent.

### 3.2.3 Glorot initialization

Since Deep Learning became popular and what is called the Deep Learning Community was given birth, different initialization strategies for the weights and the biases were proposed. We have used Glorot initialization shown below [22] to initialize the weights of the classification heads and experimented with non-pretrained image encoders that we initialized using the same strategy and fully-finetuned them. Their performance however was inferior in concept prediction.

$$\text{Glorot: } W_{i,j} \sim \mathcal{U}\left(-\sqrt{\frac{6}{f_{\text{in}} + f_{\text{out}}}}, \sqrt{\frac{6}{f_{\text{in}} + f_{\text{out}}}}\right)$$

## 3.3 Retrieval Augmented Generation

It has been impressive to Deep Learning researchers how nowadays general-purpose sequence-to-sequence models are getting really powerful, they manage to capture the world knowledge in parameters, they achieve strong results on loads of tasks and are applicable for almost everything. However, they still often hallucinate, may usually struggle to access, and apply knowledge and are difficult to update. On the other hand, modern Information Retrieval (IR) is great as well, as externally reviewed knowledge may become useful for a huge variety of NLP tasks. Modern IR provides a precise and accurate knowledge access mechanism, it is trivial to update, whereas by "modern" IR we refer to dense retrieval that starts to outperform traditional IR. On the negative side though, it still needs retrieval supervision or heuristics such as BM25, as well as some –usually task specific– way to integrate into downstream tasks. The main idea behind Retrieval Augmented Generation [45] was to combine the massive success of parametric sequence-to-sequence models in NLP with the strengths of Dense Passage Retriever (DPR) [34], which dominates modern IR and uses a FAISS index [32] that is referred to as non-parametric memory.

In the RAG approach [45], dual memory components are pre-trained and pre-loaded with extensive knowledge to encapsulate information via the representations without further training. The generator $p_\theta$ acts as a parametric memory, with the retriever $p_\eta$ embodying a non-parametric memory in the document encoder $\mathbf{d}(.)$, while including a Dense Passage Retriever (DPR) [34]. To train both the retriever $p_\eta$ and generator $p_\theta$ end-to-end, we treat the retrieved document as a latent variable $\boldsymbol{z}$, and meanwhile the embedding of the closest document representation is represented as $\mathbf{d}(\boldsymbol{z})$. The Maximum Inner Product Search (MIPS) algorithm [56] is used to compute the top $k$ retrieved documents with respect to $p_\eta(\boldsymbol{z}|\boldsymbol{x}) = \exp\left(\mathbf{d}(\boldsymbol{z})^T\mathbf{q}(\boldsymbol{x})\right)$. This type of networks that compute probabilities using the inner product of the query and document encoder embeddings are referred to in bibliography as *Siamese* or *bi-encoder* or *two towers networks*. It is also important that the document encoder is trained once, during pre-training, whereas the query encoder is trained continuously end-to-end by applying back-propagation with either Adam or AdamW optimizer that are described in section 3.2. This way, the query encoder learns how to adjust its weights to retrieve better for the downstream task it is trained for.

To conclude the generated caption $\boldsymbol{y}$ is produced by marginalizing over the predictions. The generator $p_\theta$ is a sequence-to-sequence model, a BART [44] instance precisely, which conditions on the latent documents $z$ together with each input $x$ to generate each output $\boldsymbol{y}$. As an overall component, it produces $p_\theta\left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}_{1:i-1}\right)$ to create a Language Model (LM) over the tokens vocabulary $\mathcal{V}$ given as input the latent documents $\boldsymbol{z}$ and queries $\boldsymbol{x}$. During training, we treat questions-answers as input-output pairs $(\boldsymbol{x}, \boldsymbol{y})$ and train RAG-token by directly minimizing the negative marginal log-likelihood of generating output sequences $\boldsymbol{y}$ on input sequences $\boldsymbol{x}$. If $\mathcal{D} = \{\boldsymbol{x}_j, \boldsymbol{y}_j\}_j$ is the complete dataset, our training objective is:

$$l_{\text{cross}}(\boldsymbol{x}, \boldsymbol{y}; \theta, \eta) = -\log p(\boldsymbol{y}|\boldsymbol{x}; \theta, \eta)$$

$$\sum_j l_{\text{cross}}(\boldsymbol{x}_j, \boldsymbol{y}_j; \theta, \eta) = \sum_j -\log p(\boldsymbol{y}_j|\boldsymbol{x}_j; \theta, \eta)$$

### 3.3.1 Dense Passage Retriever

Dense Passage Retriever (DPR) [34] is a dense retrieval algorithm, which dominates modern IR. Assume that our collection $\mathcal{D}$ contains $D$ documents, which are split into passages of equal lengths that correspond to basic retrieval units. The corpus takes the form $\mathcal{C} = \{p_i\}_{i=1}^{k}$, where each passage $p_i \in \mathcal{C}$ can be interpreted as a sequence of tokens $p_i = \left\{w_i^{(j)}\right\}_{j=1}^{|p_i|}$. The goal is to find a span from one of the passages that answers the question. The idea intuitively associates each of the questions to a filtered subset of the training corpus with passages that ideally answer the question. The corresponding function is $R : (q, \mathcal{C}) \rightarrow \mathcal{C}_f$ and the amount of the retrieved passages is typically smaller than the corpus size $|\mathcal{C}_f| << |\mathcal{C}|$.

The Dense Passage Retriever implementation incorporates the use of Fast AI Similarity Search (FAISS) index [32] and provides several advantages, such as GPU optimization due to improved parallelism, improved preprocessing of the document collection, use of Inverted File Index to also cope with clustering and a trainable architecture that captures better semantic similarities. Of course there are several disadvantages as well, lower performance with out of vocabulary words, higher time complexity, which means that the model is slower compared to simple heuristics as TF-IDF since it is comprised of two phases; indexing and retrieval and computational complexity, since training is facilitated by a GPU to name a few.

### 3.3.2 Generative Transformers and Denoising Autoencoders

BART [44] is a denoising autoencoder [83], which maps a corrupted document to the original one it was derived from and is used as a generator in our RAG-token pipeline that we further elaborate on in section 3.3.3. It has been implemented as a sequence-to-sequence model with a bidirectional encoder over corrupted text and a left-to-right autoregressive decoder.

Autoencoders is yet another example of unsupervised learning algorithm that maps its inputs $\boldsymbol{x}$ to some hidden space and then the hidden representations $\boldsymbol{z}$ back to the original inputs to learn a low-dimensional manifold. The key difference of denoising autoencoders compared to vanilla autoencoders is that instead of trying to reproduce the original data, we try to corrupt it with noise. Denoising autoencoders are typically overcomplete, meaning that the dimensionality of the hidden representations ought to be larger than the input dimension. The goal is to recreate from the corrupted data the original ones; so we start with an imperfect input in order to avoid the identity mapping problem, while additional regularization is not needed –corrupting the original image acts as a regularizer itself.

Precisely, to address the risk of *overfitting* when there are more network parameters than the number of data points and improve robustness, the input is partially corrupted by adding noises to or masking some values of the input vector in a stochastic manner, which is represented by a noise model $\tilde{x} \sim \mathcal{M}_{\mathcal{D}}\left(\tilde{x}|x\right)$. Then the network is trained to reconstruct the original input $x$, where the noise model $\mathcal{M}_{\mathcal{D}}$ constitutes a composition of multiple corruption processes $\mathcal{C}$. It is thus not specific to a particular type of such process and defines the mapping from the true data samples to the noisy or corrupted ones; therefore $\mathcal{M}_{\mathcal{D}} : \mathcal{D} \to \tilde{\mathcal{D}}$ such that $\tilde{x} \in \tilde{\mathcal{D}}$.

$$\begin{aligned} \tilde{x}^{(i)} &\sim \mathcal{M}_{\mathcal{D}}\left(\tilde{x}|x\right) \\ L(\theta, \phi) &= \frac{1}{n}\sum_{i=1}^{n}\left(x^{(i)} - f_{\theta}\left(g_{\phi}\left(\tilde{x}^{(i)}\right)\right)\right)^{2} \end{aligned}$$

Learning the vector field in a denoising autoencoder means attracting the data from out of the manifold to the manifold. By setting the noise model, we are trying to memorize the error, thus find the closest possible place to the manifold because of imposing an implicit constraint to the data points due to the corruption processes. We aim to find the center of mass in the manifold to attract the data. The mapping of the noisy data to the original image in the manifold basically follows the closest possible projection on it. We are learning the shape of the manifold in order to capture local similarities and that is also represented by the loss function. Far from the manifold generalization capabilities are lost. The overall ambition is to find a low-dimensional manifold of the data $\boldsymbol{x} \in \mathcal{D}$ in a high-dimensional space.

Last but not least, because of the implicit regularization imposed by the noise model, we take into account the reconstruction force to approximately recover the input data and the regularization force to further avoid the identity mapping problem. However, there are two underlying opposing forces also in the reconstruction process itself. One is the noising process that pushes us outside the manifold, what we think is important. The other is the learning process pushing us back to the manifold and these opposing forces actually shape the manifold through the reconstruction loss, which incorporates implicit regularization.

### 3.3.3 RAG token

RAG-Token is an implementation of RAG that draws a different document $\boldsymbol{z}$ to predict each target token and marginalize accordingly. In this context, the generator is allowed to combine content from several documents to produce the output caption $\boldsymbol{y}$, namely the top $k$ retrieved documents are obtained according to the scores computed by Maximum Inner Product Search algorithm [56]. To reveal the whole output sequence $\boldsymbol{y}$, we use the retriever $p_\eta$ to obtain the top $k$ documents, which we pass to the generator $p_\theta$, an encoder-decoder transformer network that is a BART instance in our case as it is described in section 3.3.2 and we marginalize per token we generate; conditioned on the previously generated ones. The corruption processes involved in BART include token masking, sentence permutation, document rotation, token deletion, text infilling and are applied to all documents considered.

$$\begin{aligned}
p(\boldsymbol{y}|\boldsymbol{x}) &= \prod_{i=1}^{N} \sum_{\boldsymbol{z}} p_\eta(\boldsymbol{z}|\boldsymbol{x}) p_\theta \left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{1:i-1}\right) \\
&= \prod_{i=1}^{N} \sum_{\boldsymbol{z}} \exp\left(\mathbf{d}(\boldsymbol{z})^T \mathbf{q}(\boldsymbol{x})\right) p_\theta \left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{1:i-1}\right)
\end{aligned}$$

The decoding part follows the typical, autoregressive sequence-to-sequence generation pipeline, with transition probabilities $p'_\theta \left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{1:i-1}\right)$ given by summing over the top $k$ representations. We plug them into a standard beam decoder that will move towards states with the highest values of these probabilities, to get an output sequence $\boldsymbol{y}$, which constitutes our model's prediction.

$$\begin{aligned}
p'_\theta \left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{1:i-1}\right) &= \sum_{\boldsymbol{z}_j} p_\eta(\boldsymbol{z}_j|\boldsymbol{x}) p_\theta \left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{1:i-1}\right) \\
&= \sum_{\boldsymbol{z}_j} \exp\left(\mathbf{d}(\boldsymbol{z}_j)^T \mathbf{q}(\boldsymbol{x})\right) p_\theta \left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{1:i-1}\right)
\end{aligned}$$

To summarize the training process, within the retriever $p_\eta$ that acts as a non parametric memory, we encode each document $\boldsymbol{x}$ into a dense representation, apply Maximum Inner Product Search and pass the documents with the maximum dot product to the sequence-to-sequence generator. Afterwards, within the generator $p_\theta$ that acts as a parametric memory, we get a prediction of the generation per document and then we marginalize to fill-in the output.

### Arithmetic stability and the logsumexp trick

In order to increase arithmetic stability of our algorithm, we use the logsumexp trick in our implementation of RAG token, which helps us avoid `NaN` values by preventing divisions with very small or very large numbers. These are considered zero or `Inf` in computers' floating-point arithmetic respectively. Given this purpose, when we aim to compute some small number $x \in \mathbb{R}$, we shall rewrite it as $x = \log e^x = \log e^{x+M-M} = \log(e^M e^{x-M}) = \log e^M + \log(e^M e^{x-M}) = M + \log(e^M e^{x-M})$, where $M \in \mathbb{R}$ takes a large value that we pick arbitarily.

Similarly, if we consider $\log \sum_{k=1}^{K} f_k$, where $f_1, f_2, ..., f_K \in \mathbb{R}$ are either extremely small or very large numbers, we may find the maximum among them $M = \max(f_1, f_2, ..., f_K)$ and then similarly apply the logsumexp trick. By computing similar steps, we derive a more stable equivalent numerical expression, therefore taking again into account that $x = \log e^x$ we may recompute it as below:

$$\log \sum_{k=1}^{K} e^{f_k} = \log \sum_{k=1}^{K} e^{f_k+M-M} = \log e^M \sum_{k=1}^{K} e^{f_k-M} = M + \log \sum_{k=1}^{K} e^{f_k-M}$$

In our implementation of RAG token, when computing the probability of the output sequence conditioned in the input $p(\boldsymbol{y}|\boldsymbol{x})$, in order to reveal the whole output sequence $\boldsymbol{y}$, as we marginalize per token we generate, consuming the previously generated ones; applying the logsumexp trick is based on:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{x}) &= \prod_{i=1}^{N} \sum_{\boldsymbol{z}} p_\eta(\boldsymbol{z}|\boldsymbol{x}) p_\theta\left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{1:i-1}\right) \\ &\propto \prod_{i=1}^{N} \underbrace{\sum_{\boldsymbol{z}} \mathcal{L}_\eta\left(\boldsymbol{z}|\boldsymbol{x}\right) + \mathcal{L}_\theta\left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{1:i-1}\right)}_{\text{sum of log probabilities}} \\ &\propto \sum_{i=1}^{N} \log\left(\sum_{\boldsymbol{z}} p_\eta(\boldsymbol{z}|\boldsymbol{x}) + p_\theta\left(\boldsymbol{y}_i|\boldsymbol{x}, \boldsymbol{y}_{1:i-1}\right)\right) \end{aligned}$$

$$\text{according to: } \sum_{\boldsymbol{z},\boldsymbol{w}\in\{\eta,\theta\}} \mathcal{L}_{\boldsymbol{w}}(.) = \log \sum_{\boldsymbol{z},\boldsymbol{w}\in\{\eta,\theta\}} \exp\left(\mathcal{L}_{\boldsymbol{w}}(.)\right) = \underbrace{\log \sum_{\boldsymbol{z},\boldsymbol{w}\in\{\eta,\theta\}} P_{\boldsymbol{w}}(.)}_{\text{log of summed probabilities}}$$

## 3.4 Evaluation goals and delimitations

Evaluation includes mainly the metrics which constitute part of `pycocoevalcap` library for Interactive Python in Unix systems, including BLEU [57], ROUGE [49], METEOR [6], CiDER [82] and SPICE [3]. They are typically used for the evaluation of captioning systems and thus allow us to quantitatively compare our approach to existing baselines and more novel algorithms.

A selection among these metrics is used to evaluate submissions in ImageCLEF medical 2022 [68] as well that precisely include BLEU 1-4 scores for caption generation and F1 score for concepts' prediction. The default implementation[1] of F1 score from scikit-learn is computed for each image and then all scores are summed and averaged over all images. When using F1 score, evaluation is conducted in terms of set coverage metrics such as precision, recall, and combinations thereof. Moreover, all the scores for caption evaluation, such as NIST, METEOR, ROUGE, and Word Error Rate (WER), have been designed to be robust to variability in style and wording of the generated captions. This robustness does not imply perfection though.

Interestingly, the aforementioned evaluation measures employed by DC research mainly assess lexical overlap between machine-generated and human-authored gold captions, without directly assessing clinical correctness. As it is indicated in the study in [59] this can lead to cases where a clinically wrong generated report can be scored higher than a clinically correct one, for instance "*pneumothorax* would be considered a positive find in *no pneumothorax is observed*" [93], which apart from being clinically incorrect, it also does not make sense. In addition, ROUGE metrics mainly consider the content by measuring *n*-gram overlaps within the text and not its readability, which may lead to poor grammar evaluation [33].

Furthermore, although higher quantitative accuracy is most often better, there are categorical differences of the DC methods as well, which relate to qualitative evaluation of our approach and may refer to its practical usefulness. However, it is not yet obvious what metrics can be used in order to obtain practical information about the quality of the generated captions, which is left for future work. Our ambition from developing and training Deep Networks based on the transformer architecture [81] is to improve performance of Encoder-Decoder models with complex cells [48, 84], traditionally achieving state of the art results.

---

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

## 3.5 Significance tests for experiments outcomes

In addition, we focus on pairwise variance analysis for the models. The F-distribution is primarily used to compare the variances of two populations, which is particularly relevant in analysis of variance testing and regression analysis. The F-distribution with $n_1$, $n_2$ degrees of freedom is defined by:

$$F(n_1, n_2) = \frac{\frac{X^2(n_1)}{n_1}}{\frac{X^2(n_2)}{n_2}}$$

Using this definition, we can test whether the variances of two populations are equal, which has been performed and illustrated below. In order to deal exclusively with the right tail of the distribution, when taking ratios of sample variances from the theorem we should put the largest variance in the numerator of $s_1^2/s_2^2 \sim F(n_1 - 1.n_2 - 1)$. In particular, we perform two-tailed F-tests comparing the variances of the samples in ranges of the different models results to show the two-tailed probability that the variance of the data in the respective ranges are significantly different. However, this is not though always the case.

What is more, we perform the Wilcoxon Rank-Sum test, as well as the Mann-Whitney U–test that is essentially an alternative form of the former for independent samples. We perform both a Wilcoxon Rank-Sum test and a Mann-Whitney U–test for Paired Samples again using a predefined significance level $5 \times 10^{-2}$ to test the following null hypothesis $H_0$: any differences between the two models is due to chance (based on the median of the differences). Two refers to the fact that we perform pairwise tests.

Moreover, the intuition behind using the Kolmogorov-Smirnov test apart from one-tailed and two-tailed statistical significance T-tests shown above has been that there might be measurements where the population means of the score distributions are similar but either there is a remarkable difference in variances or one distribution is bimodal that only the Kolmogorov-Smirnov test would detect. Additionally, although the results might verify the statistical significance, the opposite could happen as well. In that case, we additionally want to assess the effect size [75] to measure how large a difference is after getting a failed hypothesis test. For this purpose, we used Cohen's $d$ and consider the effect size to be small when $d < 0.2$, medium when $0.2 \leq d \leq 0.5$, large when $d > 0.5$ as vastly in literature.

# Chapter 4

# Model selection and hyperparameter tuning

## 4.1 ImageCLEFmedical 2022 contributions

In this part, we describe the core components of the methods utilized to encode the X-RAYs with dense embeddings and explain in detail the baseline networks that we proposed in ImageCLEFmedical 2022 evaluation campaign, in order of performance, for both subtasks that are based on the aforementioned core components that rely on pre-trained architectures, which are extremely popular in computer vision. Our group, *NeuralDynamicsLab* at KTH Royal Institute of Technology, within the school of Electrical Engineering and Computer Science, ranked $4^{\text{th}}$ in the former and $5^{\text{th}}$ in the latter task [55] among 12 best research groups.

Precisely, we provide details about the ImageCLEFmedical 2022 concept detection and caption prediction datasets and on how we designed backbone networks as generic image encoders that rely on Convolutional Neural Networks (CNN) architectures. They are popular for vision tasks on generic images, such as classification and semantic segmentation, while they are shared within all baselines, in both ImageCLEFmedical Caption tasks. Furthermore, we analyze all the components of each submission as well as give details regarding hyper-parameter tuning.

For all our models, we have set in advance all the random seeds equal to $0$, the CUDNNs backends as deterministic and disabled the CUDNNs backends benchmark to ensure consistency of the aforementioned splits in consecutive runs for hyper-parameter selection. This procedure has been applied during pre-processing for both subtasks of the evaluation campaign.

### 4.1.1 Concept prediction subtask

As we mentioned in section 1.5 “backbone networks” refer to image encoders, which are state-of-the-art architectures, pretrained on ImageNet classification dataset [70], shared for both subtasks. In the case of concept prediction, an additional classification head that is either a Perceptron or a Multi-layered Perceptron was added on top of these “backbone networks” and its weights were initialized using Glorot initialization strategy [22] as we previously described.

#### a. Pre-trained DenseNet161 with fine-tuned classification head, learning rate $10^{-3}$, Adam optimizer and gradient clipping

The first two models correspond to a DenseNet161 convolutional network, which is pretrained on ImageNet classification dataset, while its head is a Perceptron that is further fine-tuned on the ImageCLEFmedical 2022 dataset using sigmoid activation function in the output units that equal the number of concepts -therefore $8374$ nodes, a constant learning rate equal to $10^{-3}$ and the negative $F_1$ score as a minimization criterion. For each image, we assign it the concepts that have predicted probabilities above $50\%$, while the tags obtain their numerical IDs in their order of appearance before shuffling them. Furthermore, we clip the gradients computed during training to be in $[-1, 1]$, to increase numerical stability.

We have used the Adam optimizer [37], which is described in section 3.2.1, so that progress along steep directions is damped and meanwhile progress along flat directions is accelerated. Our best performing baseline is an instance of the aforementioned architecture trained in all the provided data, thus after merging again the training, validation and development sets that are described in section 3.1.1 and achieves $F_1 = 0.43601$[1]. The next model corresponds to the same network architecture but is trained only in training set and achieves a score $F_1 = 0.43567$. For the latter case, where we have measured performance in all sets, we present plots with the evolution of $F_1$ score and accuracy during training in Figure 4.1.1(a).

---

[1] All performance scores reported in this chapter correspond to the test set. For a detailed analysis of the model results in the training, validation, development and test sets; please refer to chapter 6 where statistical significance tests are also included.

### b. Pre-trained DenseNet161 with fine-tuned classification head, learning rate $5 \times 10^{-4}$, AdamW optimizer and gradient clipping

The next model corresponds to another DenseNet161 convolutional network that is pretrained on ImageNet classification dataset and its head is again a Perceptron that is further fine-tuned on the ImageCLEFmedical 2022 data using sigmoid activation function in the output units that equal the number of concepts -thus $8374$ nodes, a constant learning rate equal to $5 \times 10^{-4}$ and the negative $F_1$ score as a minimization criterion. For each image, we assign it the concepts that have predicted probabilities above $50\%$, while the tags obtain their numerical IDs in their order of appearance before shuffling them. Furthermore, we clip the gradients computed during training to be in $[-1, 1]$, to ensure numerical stability.

In this occasion we have used an improved version of Adam optimizer, called AdamW, which is described in section 3.2.2. Our model is an instance of the aforementioned network architecture, it is trained only in training set and achieves a score $F_1 = 0.43558$, although we would expect training with AdamW to perform better. Since the gain of re-training the model after merging all the splits is almost negligible, as we already noticed in section 4.1.1, the remaining models are not re-trained in the entire dataset. Once again, we present plots with the evolution of $F_1$ score and accuracy in Figure 4.1.1(b) in order to compare the different configurations.

### c. Pre-trained DenseNet161 with fine-tuned classification head, learning rate $5 \times 10^{-4}$ and Adam optimizer

The subsequent model is yet another DenseNet161 convolutional network, which is pretrained on ImageNet classification dataset and its head is another Perceptron that is further fine-tuned on the ImageCLEFmedical 2022 data using sigmoid activation function in the output units that equal the number of concepts -thus $8374$ nodes, a constant learning rate equal to $5 \times 10^{-4}$ and the negative $F_1$ score as a minimization criterion. For each image, we assign it the concepts that have predicted probabilities above $50\%$, while the tags obtain their numerical IDs in their order of appearance before shuffling them and train the network using the Adam optimizer; as we have excessively described in section 3.2.1.

Our model is an instance of the aforementioned network architecture and achieves a score $F_1 = 0.43539$, however, in this baseline we omit clipping the gradients, in contrast with the models described above in sections 4.1.1(a) and 4.1.1(b). Furthermore, as for both previous best-performing models we present plots with the evolution of $F_1$ score and accuracy below in Figure 4.1.1(c).

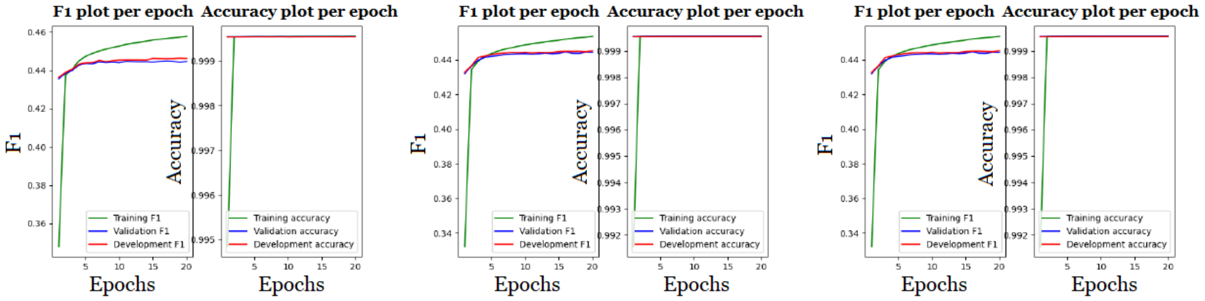


Figure 4.1.1: $F_1$ and accuracy train, val., dev. scores plots per epoch for the models (a) of section 4.1.1(a), (b) of section 4.1.1(b), as well as (c) of section 4.1.1(c). We observe that the classifications heads, which we finetuned on ImageCLEFmedical 2022 data, appear to be sufficiently regularized (thus there is no overfitting) and to have used their maximum capacity.

### d. Ensemble of pre-trained DenseNet CNNs with fine-tuned classification heads

The proceeding model and the best performing mixture of individual weak learners corresponds to the 10 best performing DenseNet CNNs, including instances of both DenseNet161 and DenseNet121 architectures, and indicates our quest for diversity and to consequently exploit the “Wisdom of the crowd” [76]; although their performance was lower compared to previous models.

In this context, we have taken into account the “votes” of all the different CNNs to make decisions on the assigned tags. The voting scheme consists of averaging the probabilities computed by the different weak learners before assigning to each image the concepts that have average predicted probabilities above $50\%$, while the tags as usual obtain their numerical IDs in their order of appearance before shuffling them. We also experimented with using alternative voting policies, such as computing the union or intersection of the assigned tags by each weak learner, where assignments are defined by the predicted probabilities being above $50\%$, in the pool of finetuned networks, but they performed poorly.

Table 4.1.1 summarizes the architecture of all individual networks in the pool of encoders. This precisely includes the type of Backbone Network, the optimizer, the value of learning rate and whether it is decaying per epoch, as well as their batch size.

Note that for all weak learners in this pool of encoders, the classification head is always a Perceptron, which is further fine-tuned in the ImageCLEFmedical 2022 data using sigmoid activation function in the output units that equal the number of concepts. Moreover, when linear decay is applied, the learning rate is updated by: $\eta_{t+1} = \eta_0 \times \frac{1-t}{T}$ where $t$ represents the current time step, $T$ the total number of epochs and $\eta_0$ is the learning rate at the beginning of training procedure. The performance of this mixture of experts equals $F_1 = 0.43496$ on the test set.

Table 4.1.1: Summary of weak learners' architecture and training regime in ensemble

| Backbone Net. | Optimizer | Learning Rate | Batch size | Epochs | Other Remarks |
|---|---|---|---|---|---|
| 1. DenseNet121 | AdamW | $5 \times 10^{-4}$ | 60 | 20 | - |
| 2. DenseNet121 | AdamW | $10^{-3}$ | 60 | 20 | - |
| 3. DenseNet121 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 4. DenseNet161 | Adam | $10^{-3}$ | 120 | 20 | - |
| 5. DenseNet161 | AdamW | $10^{-3}$ | 120 | 20 | Linear Decay |
| 6. DenseNet161 | Adam | $5 \times 10^{-4}$ | 120 | 20 | - |
| 7. DenseNet161 | Adam | $5 \times 10^{-4}$ | 120 | 20 | No Grad. Clipping |
| 8. DenseNet161 | AdamW | $5 \times 10^{-4}$ | 120 | 20 | - |
| 9. DenseNet161 | AdamW | $10^{-4}$ | 120 | 50 | - |
| 10. DenseNet161 | AdamW | $10^{-4}$ | 120 | 20 | - |

### e. Ensemble of various pre-trained CNNs with fine-tuned classification heads

Although Dense Convolutional Networks (DenseNet CNNs) appear to outperform other network architectures, which is in line with their extensive use in biomedical applications that include X-RAYs processing [64], we experimented with a plethora of CNNs backbone networks as we have mentioned in section 1.5. Consequently, the ensuing three models constitute ensembles that include different architectures within their members, with varying hyperparameter values to encourage diversity of training regimes. During the voting process we average the probabilities computed by the softmax layer of all different week learners before assigning to each image the tags that have average predicted probabilities above $50\%$.

Our three following mixtures of experts achieve a score $F_{1,1} = 0.43404$, $F_{1,2} = 0.43130$, $F_{1,3} = 0.42957$ respectively. Tables 4.1.2, 4.1.3, 4.1.4 summarize the architecture of all individual networks in each pool of encoders. Their format is identical to that used in section 4.1.1(d) and consequently they also refer to the hyper-parameter values for each of the weak learners.

Note that the classification head is always a Perceptron which is further fine-tuned in the ImageCLEFmedical 2022 data using sigmoid activation function in the output units that equal the number of concepts. Moreover, when linear decay is applied, the learning rate is updated by: $\eta_{t+1} = \eta_0 \times \frac{1-t}{T}$ where $t$ represents the current time step, $T$ the total number of epochs and $\eta_0$ is the initial learning rate.

Table 4.1.2: Summary of weak learners' architecture and training regime in ensemble

| Backbone Net. | Optimizer | Learning Rate | Batch size | Epochs | Other Remarks |
|---|---|---|---|---|---|
| 1. AlexNet | AdamW | $10^{-4}$ | 60 | 20 | - |
| 2. AlexNet | AdamW | $5 \times 10^{-5}$ | 60 | 20 | - |
| 3. DenseNet121 | AdamW | $5 \times 10^{-4}$ | 60 | 20 | - |
| 4. DenseNet121 | AdamW | $10^{-3}$ | 60 | 20 | - |
| 5. DenseNet121 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 6. DenseNet161 | Adam | $10^{-3}$ | 120 | 20 | - |
| 7. DenseNet161 | AdamW | $10^{-3}$ | 120 | 20 | Linear Decay |
| 8. DenseNet161 | Adam | $5 \times 10^{-4}$ | 120 | 20 | - |
| 9. DenseNet161 | Adam | $5 \times 10^{-4}$ | 120 | 20 | No Grad. Clipping |
| 10. DenseNet161 | AdamW | $5 \times 10^{-4}$ | 120 | 20 | - |
| 11. ResNet50 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 12. ResNet101 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 13. VGG-13 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 14. VGG-16 | AdamW | $10^{-4}$ | 60 | 20 | - |

Table 4.1.3: Summary of weak learners' architecture and training regime in ensemble

| Backbone Net. | Optimizer | Learning Rate | Batch size | Epochs | Other Remarks |
|---|---|---|---|---|---|
| 1. AlexNet | AdamW | $10^{-4}$ | 60 | 20 | - |
| 2. AlexNet | AdamW | $5 \times 10^{-5}$ | 60 | 20 | - |
| 3. DenseNet121 | AdamW | $5 \times 10^{-4}$ | 60 | 20 | - |
| 4. DenseNet121 | AdamW | $10^{-3}$ | 60 | 20 | - |
| 5. DenseNet161 | Adam | $10^{-3}$ | 120 | 20 | - |
| 6. DenseNet161 | AdamW | $10^{-3}$ | 120 | 20 | Linear Decay |
| 7. ResNet50 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 8. ResNet50 | AdamW | $5 \times 10^{-5}$ | 60 | 20 | - |
| 9. ResNet101 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 10. ResNet101 | AdamW | $5 \times 10^{-4}$ | 60 | 20 | - |
| 11. VGG-13 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 12. VGG-13 | AdamW | $5 \times 10^{-5}$ | 60 | 20 | - |
| 13. VGG-16 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 14. VGG-16 | AdamW | $5 \times 10^{-5}$ | 60 | 20 | - |

Table 4.1.4: Summary of weak learners' architecture and training regime in ensemble

| Backbone Net. | Optimizer | Learning Rate | Batch size | Epochs | Other Remarks |
|---|---|---|---|---|---|
| 1. AlexNet | AdamW | $10^{-4}$ | 60 | 20 | - |
| 2. AlexNet | AdamW | $5 \times 10^{-5}$ | 60 | 20 | - |
| 3. DenseNet121 | AdamW | $5 \times 10^{-4}$ | 60 | 20 | - |
| 4. DenseNet121 | AdamW | $10^{-3}$ | 60 | 20 | - |
| 5. DenseNet121 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 6. DenseNet161 | Adam | $10^{-3}$ | 120 | 20 | - |
| 7. DenseNet161 | AdamW | $10^{-3}$ | 120 | 20 | Linear Decay |
| 8. DenseNet161 | Adam | $5 \times 10^{-4}$ | 120 | 20 | - |
| 9. ResNet50 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 10. ResNet101 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 11. VGG-13 | AdamW | $10^{-4}$ | 60 | 20 | - |
| 12. VGG-16 | AdamW | $10^{-4}$ | 60 | 20 | - |

### f. Fully fine-tuned DenseNet161 with cyclical learning rate and AdamW optimizer

The succeeding model corresponds to a DenseNet161 convolutional network that is now fully-finetuned on the ImageCLEFmedical 2022 data using sigmoid activation function in the output units that equal the number of concepts -thus $8374$ nodes, scheduled learning rate [39] and the negative $F_1$ score as a minimization criterion. For each image, we assign it the concepts that have predicted probabilities above $50\%$, while the tags obtain their numerical IDs in their order of appearance before shuffling them as in all previous pipelines.

One important aspect of minibatch or stochastic gradient descent relates to the choice of the learning rate $\eta$ that controls the size of the update, which will occur to the gradients in every iteration. Constant learning rates have been traditionally used to train Deep Neural Networks based on back-propagation algorithm, although do not guarantee optimal convergence rate according to the Stochastic Approximation Theory [67], precisely the network parameters hover around a minimum at an average distance proportional to the learning rate and to a variance that is dependent on the objective function and the exemplar set [13]. To this end, cyclical learning rates have been proposed as a new method for setting the learning rate by cyclically varying its value between reasonable boundary values, which increases classification accuracy when training CNNs with generic images [74].

A high value of $\eta$ will make the network make large steps above the minimum of the

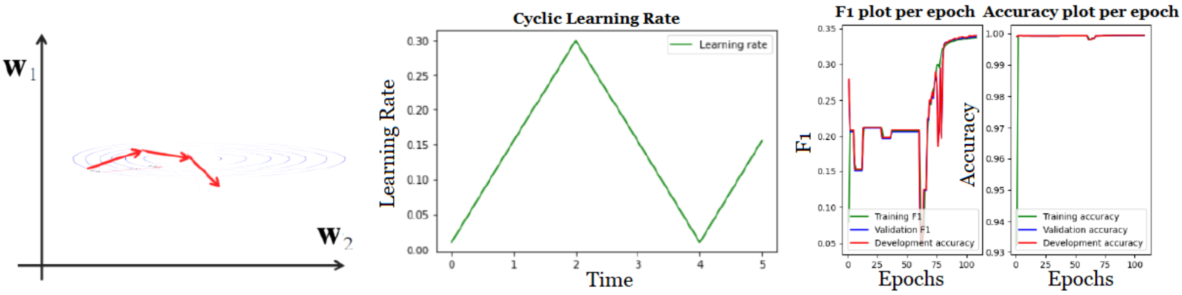


Figure 4.1.2: (a) Schematic illustration of the error landscape with a high learning rate, (b) example plot of a cyclical learning rate with $\eta_{\text{min}} = 0.01$, $\eta_{\text{max}} = 0.30$, $n_s = 2$ and (c) $F_1$ and accuracy train, val., dev. scores plots per epoch for the model of section 4.1.1.

error function but never converge to it, as illustrated in Figure 4.1.2(a). A small value of $\eta$ will delay convergence, preventing the network to find a minimum of the error function if the number of epochs is limited. A cyclical learning rate linearly ranges between two values $\eta_{\text{min}}$ and $\eta_{\text{max}}$. One maximization of the learning rate followed by a minimization is called a cycle. In Figure 4.1.2(b) we present an indicative example of cyclical learning rate, where $\eta_{\text{min}} = 0.01$, $\eta_{\text{max}} = 0.30$, $n_s = 2$ and we denote as $2n_s$ the time required for a cycle of our learning rate to complete. In our model we set $\eta_{\text{min}} = 10^{-5}$, $\eta_{\text{max}} = 0.1$, $n_s = 4$ for the first $80$ epochs and then set it to a constant value $\eta = 10^{-3}$ for $30$ additional epochs.

This network achieves a score $F_1 = 0.31687$, which is a rather lower score compared to the pre-trained models on ImageNet classification dataset [70], achieving more than $10\%$ higher $F_1$ results on the test set. Moreover, we present plots with the evolution of $F_1$ score and accuracy per training epoch of the model in Figure 4.1.2(c) that is quite unstable while varying the learning rate.

### g. Nearest Neighbours Baseline

The ensuing model is a generalization of the 1-NN baseline proposed in [50]. We further either remind or inform the reader that for every image in the test set, the 1-NN baseline assigns the tags of the visually most similar image from the training set as the output and consequently for every image, $\hat{x}$, in the test set, the 1-NN baseline will output the set of concepts, say $y^*$, of the most similar image, say $x^*$, from the training set as output [59]. Therefore, if we denote by $\mathbf{e}(.)$ the output of the employed image encoder among those mentioned in section 1.5, 1-NN predicts $(\hat{x}, \hat{y}) = (\hat{x}, y^*)$ that satisfies $(x^*, y^*) = \arg\min_{\hat{x}} \cos\left(\mathbf{e}(\hat{x}), \mathbf{e}(x^*)\right)$. Our generalized Nearest Neighbours baseline takes into account $k \in \mathbb{Z}^+$ neighbours instead and not necessarily only the one with closest representation. Our best performing $k$-NN model though uses $k = 1$ with a VGG-16 encoder pre-trained on ImageNet classification dataset and achieves only $F_1 = 0.25061$ that indicates the importance of fine-tuning, which is impossible to conduct in this baseline.

### 4.1.2 Caption generation subtask

In ImageCLEFmedical 2022 evaluation campaign, “the first step to automatic image captioning and scene understanding boils down to identifying the presence and location of relevant concepts within a large corpus of medical images” that is followed by caption generation in captioning. Based on medical images content, the concept prediction task provides the building blocks for scene understanding by identifying the individual components, referred to as image tags, from which captions are composed. The assigned concepts can be further applied for context-based image and information retrieval purposes” [68].

“On the basis of the vocabulary $\mathcal{V}$ identified during concept prediction task, as well as the visual information of their interaction in the image, caption generation task refers to composing coherent captions for each entire image. For the medical captioning task, rather than the mere coverage of visual concepts, detecting the interplay of visible elements can be crucial for strong performance” [68]. In the following, we describe our proposed models for Diagnostic Captioning, in which the generalized Nearest Neighbours baseline that we introduced in section 4.1.1(g) has a crucial role despite it performing poorly as is.

#### a. $(1 + k)$-NN image retriever with Pegasus summarizer

Our best performing captioning models extend the Nearest Neighbours baseline for caption generation. Precisely, the 1-NN [50] constitutes one of the model components, where for every image in the test set, it will produce the diagnostic text of the visually most similar image from the training set as the output and consequently it will assign the corresponding caption, say $y^*$, of the most similar image, say $x^*$, from the training set as output [59]. Thus, if we denote by $\mathbf{e}(.)$ the output of the employed image encoder among those mentioned in section 1.5, 1-NN predicts $(\hat{x}, \hat{y}) = (\hat{x}, y^*)$ that satisfies $(x^*, y^*) = \arg\min_{\hat{x}} \cos(\mathbf{e}(\hat{x}), \mathbf{e}(x^*))$. This prediction constitutes the first part of the models’ generated caption.

In our proposed generalized baseline however, apart from the neighbour with the closest representation, we retrieve the top-$(k + 1)$ nearest neighbours, concatenate their outputs, excluding that of the most similar image and feed them as input to an abstractive summarizer; Pegasus [91] that is based on the transformer architecture, one idea that revolutionized Natural Language Processing [81] and it is trained with a Masked Language Modelling objective, which became popular within the research community though BERT [15].

For our models we employed a pre-trained AlexNet CNN on ImageNet classification dataset as our image encoder and merged our training, validation and development sets that are described in section 3.1.1, in order to benefit from an extensive set of training data to compute similarities with the test data. For each of them we keep the caption of the visually most similar image, concatenate the captions of the $k$ proceeding ones and give them as input to Pegasus summarizer, which we allow to produce a summary of maximum length $n$ tokens to eliminate repetitions. We exclude phrases as “All images are copyrighted.” and “Images courtesy of AFP, EPA, Getty” that were probably included in Pegasus’ training set from our generated summaries. The predicted captions constitute the concatenation of 1-NN baseline and Pegasus summarizer outputs. Table 4.1.5 below presents all configurations’ hyper-parameter values, namely the neighbours amount $k$ and summary length $n$, as well as their BLEU scores in decreasing order in the test set [57].

Table 4.1.5: Summary of our configurations’ hyper-parameters and statistics

| Backbone Network | Captions $k$ | Tokens $n$ | BLEU scores |
|---|---|---|---|
| AlexNet | $k = 9$ | $n = 15$ | 0.29166 |
| AlexNet | $k = 4$ | $n = 15$ | 0.28343 |
| AlexNet | $k = 3$ | $n = 15$ | 0.27855 |
| AlexNet | $k = 2$ | $n = 15$ | 0.27007 |
| AlexNet | $k = 4$ | $n = 5$ | 0.25521 |
| AlexNet | $k = 3$ | $n = 5$ | 0.25334 |

#### b. $k$-NN image retriever with Retrieval Augmented Generation

The goal of Retrieval Augmented Generation (RAG) [45], which has been excessively described in section 3.3 and used as model component, pretrained on Wikipedia with a FAISS index [32] built on $42\%$ of PubMed 2022 including recent publications related to the fields of neuroscience and computational biology; is to combine the strengths of sequence-to-sequence models and explicit knowledge retrieval. Obviously, RAG is also blended with the 1-NN baseline; namely its outputs are concatenated with the caption of the visually most similar image from the training set to produce caption predictions. This model uses either a pre-trained AlexNet or VGG-16 CNN on ImageNet classification dataset as backbone network and, despite it containing a non-parametric memory, additional to storing information in the parameters of a sequence-to-sequence generative model that is a Bidirectional Auto-Regressive Transformers (BART) generator [44], after merging our training, validation and development sets that are described in section 3.1.1 to take advantage of more input-output pairs $(\boldsymbol{x}, \boldsymbol{y})$, achieves a lower BLEU score than its predecessors described in 4.1.2(a) and according to Table 4.1.6 below. These results could possibly improve if we store extracts from patients' previous diagnoses instead of the biomedical articles or use a domain-related generative model, which is left for future work.

Table 4.1.6: Summary of our configurations' image encoders and statistics

| Backbone Network | Captions $k$ | BLEU scores |
|---|---|---|
| AlexNet | $k = 1$ | $0.25127$ |
| VGG-16 | $k = 1$ | $0.23958$ |

### c. 1-NN image retrieval baseline

Last but not least, we attempted using the 1-NN baseline [59] as is to generate the diagnostic text within the captions, which however achieved a lower score than all the aforementioned approaches. Although at first, one could interpret this as RAG models examined in section 3.3, perform better than solely the 1-NN baseline; when the latter is combined with abstractive summarization techniques for the diagnostic texts of $k$ additional visually similar images from the training set, where $k \in \mathbb{Z}^+$, it may perform better as it is indicated in section 4.1.2. Our models use a pre-trained AlexNet or VGG-16 CNN on ImageNet classification dataset as image encoder, our training, validation and development sets that are described in section 3.1.1 merged together and achieve a BLEU score according to Table 4.1.7.

Table 4.1.7: Summary of our configurations' image encoders and statistics

| Backbone Network | Captions $k$ | BLEU scores |
|---|---|---|
| AlexNet | $k = 1$ | $0.24064$ |
| VGG-16 | $k = 1$ | $0.22757$ |

### 4.1.3 Concept Detection Performance summary

Table 4.1.8 below summarizes several characteristics of the proposed baselines for the concept detection task, in order of performance with respect to $F_1$ scores. We observe that DenseNet161 encoders with finetuned classification heads are our top performing configurations and outperform other CNNs. Additional details and $F_1$ scores are provided in chapters 6 and 7 (Appendices).

Table 4.1.8: Summary of our configurations' training targets and $F_1$ scores

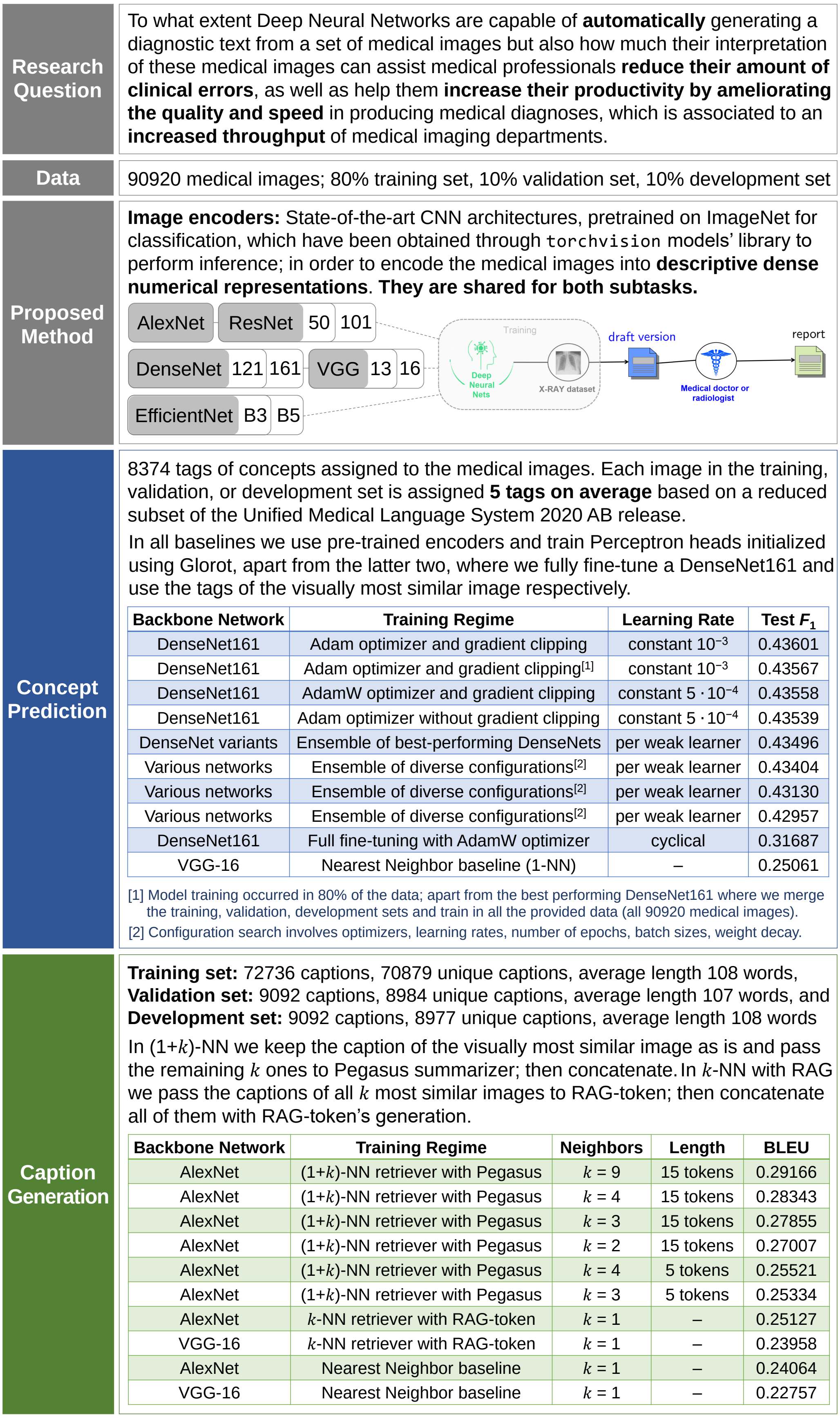
| Section | Table | Training target | Dev. $F_1$ | Val. $F_1$ | Test $F_1$ |
|---|---|---|---|---|---|
| Section 4.1.1(a) | - | DenseNet161 Head | 0.44460 | 0.44614 | 0.43601 |
| Section 4.1.1(a) | - | DenseNet161 Head | 0.44460 | 0.44614 | 0.43567 |
| Section 4.1.1(b) | - | DenseNet161 Head | 0.44429 | 0.44516 | 0.43558 |
| Section 4.1.1(c) | - | DenseNet161 Head | 0.44430 | 0.44524 | 0.43539 |
| Section 4.1.1(d) | Table 4.1.1 | Ensemble of DenseNets | 0.44544 | 0.44553 | 0.43496 |
| Section 4.1.1(e) | Table 4.1.2 | Ensemble of Networks | 0.44170 | 0.44167 | 0.43404 |
| Section 4.1.1(e) | Table 4.1.3 | Ensemble of Networks | 0.44305 | 0.44379 | 0.43130 |
| Section 4.1.1(e) | Table 4.1.4 | Ensemble of Networks | 0.44543 | 0.44623 | 0.42957 |
| Section 4.1.1(f) | - | DenseNet161 (finetuned) | 0.32418 | 0.32654 | 0.31687 |
| Section 4.1.1(g) | - | VGG-16 NN search | 0.25202 | 0.25276 | 0.25061 |

### 4.1.4 Caption Generation Performance summary

Table 4.1.9 below illustrates several characteristics and geometric mean of BLEU 1-4 scores for our caption generation baselines, in order of performance with respect to test scores. Although RAG models perform better than solely the 1-NN baseline, if the latter is combined with abstractive summarization techniques for the diagnostic texts, it is capable of performing better.

Table 4.1.9: Summary of our configurations' parameters and IDs

| Table | Encoder | Generator | Captions $k$ | Tokens $n$ | Val. score | Test score |
|---|---|---|---|---|---|---|
| Table 4.1.5 | AlexNet | Pegasus | $k = 9$ | $n = 15$ | 0.14800 | 0.29166 |
| Table 4.1.5 | AlexNet | Pegasus | $k = 4$ | $n = 15$ | 0.15500 | 0.28343 |
| Table 4.1.5 | AlexNet | Pegasus | $k = 3$ | $n = 15$ | 0.15700 | 0.27855 |
| Table 4.1.5 | AlexNet | Pegasus | $k = 2$ | $n = 15$ | 0.15600 | 0.27007 |
| Table 4.1.5 | AlexNet | Pegasus | $k = 4$ | $n = 5$ | 0.15800 | 0.25521 |
| Table 4.1.5 | AlexNet | Pegasus | $k = 3$ | $n = 5$ | 0.15600 | 0.25334 |
| Table 4.1.6 | AlexNet | RAG | $k = 1$ | - | 0.17000 | 0.25127 |
| Table 4.1.6 | VGG-16 | RAG | $k = 1$ | - | 0.19300 | 0.23958 |
| Table 4.1.7 | AlexNet | 1-NN | $k = 1$ | - | 0.15600 | 0.24064 |
| Table 4.1.7 | VGG-16 | 1-NN | $k = 1$ | - | 0.14400 | 0.22757 |



**Research Question**

To what extent Deep Neural Networks are capable of **automatically** generating a diagnostic text from a set of medical images but also how much their interpretation of these medical images can assist medical professionals **reduce their amount of clinical errors**, as well as help them **increase their productivity by ameliorating the quality and speed** in producing medical diagnoses, which is associated to an **increased throughput** of medical imaging departments.

**Data**

90920 medical images; 80% training set, 10% validation set, 10% development set

**Proposed Method**

**Image encoders:** State-of-the-art CNN architectures, pretrained on ImageNet for classification, which have been obtained through `torchvision` models' library to perform inference; in order to encode the medical images into **descriptive dense numerical representations**. **They are shared for both subtasks.**

**Concept Prediction**

8374 tags of concepts assigned to the medical images. Each image in the training, validation, or development set is assigned **5 tags on average** based on a reduced subset of the Unified Medical Language System 2020 AB release.

In all baselines we use pre-trained encoders and train Perceptron heads initialized using Glorot, apart from the latter two, where we fully fine-tune a DenseNet161 and use the tags of the visually most similar image respectively.

| Backbone Network | Training Regime | Learning Rate | Test $F_1$ |
|---|---|---|---|
| DenseNet161 | Adam optimizer and gradient clipping | constant $10^{-3}$ | 0.43601 |
| DenseNet161 | Adam optimizer and gradient clipping[1] | constant $10^{-3}$ | 0.43567 |
| DenseNet161 | AdamW optimizer and gradient clipping | constant $5 \cdot 10^{-4}$ | 0.43558 |
| DenseNet161 | Adam optimizer without gradient clipping | constant $5 \cdot 10^{-4}$ | 0.43539 |
| DenseNet variants | Ensemble of best-performing DenseNets | per weak learner | 0.43496 |
| Various networks | Ensemble of diverse configurations[2] | per weak learner | 0.43404 |
| Various networks | Ensemble of diverse configurations[2] | per weak learner | 0.43130 |
| Various networks | Ensemble of diverse configurations[2] | per weak learner | 0.42957 |
| DenseNet161 | Full fine-tuning with AdamW optimizer | cyclical | 0.31687 |
| VGG-16 | Nearest Neighbor baseline (1-NN) | – | 0.25061 |

[1] Model training occurred in 80% of the data; apart from the best performing DenseNet161 where we merge the training, validation, development sets and train in all the provided data (all 90920 medical images).

[2] Configuration search involves optimizers, learning rates, number of epochs, batch sizes, weight decay.

**Caption Generation**

**Training set:** 72736 captions, 70879 unique captions, average length 108 words,
**Validation set:** 9092 captions, 8984 unique captions, average length 107 words, and
**Development set:** 9092 captions, 8977 unique captions, average length 108 words

In (1+$k$)-NN we keep the caption of the visually most similar image as is and pass the remaining $k$ ones to Pegasus summarizer; then concatenate. In $k$-NN with RAG we pass the captions of all $k$ most similar images to RAG-token; then concatenate all of them with RAG-token's generation.

| Backbone Network | Training Regime | Neighbors | Length | BLEU |
|---|---|---|---|---|
| AlexNet | (1+$k$)-NN retriever with Pegasus | $k = 9$ | 15 tokens | 0.29166 |
| AlexNet | (1+$k$)-NN retriever with Pegasus | $k = 4$ | 15 tokens | 0.28343 |
| AlexNet | (1+$k$)-NN retriever with Pegasus | $k = 3$ | 15 tokens | 0.27855 |
| AlexNet | (1+$k$)-NN retriever with Pegasus | $k = 2$ | 15 tokens | 0.27007 |
| AlexNet | (1+$k$)-NN retriever with Pegasus | $k = 4$ | 5 tokens | 0.25521 |
| AlexNet | (1+$k$)-NN retriever with Pegasus | $k = 3$ | 5 tokens | 0.25334 |
| AlexNet | $k$-NN retriever with RAG-token | $k = 1$ | – | 0.25127 |
| VGG-16 | $k$-NN retriever with RAG-token | $k = 1$ | – | 0.23958 |
| AlexNet | Nearest Neighbor baseline | $k = 1$ | – | 0.24064 |
| VGG-16 | Nearest Neighbor baseline | $k = 1$ | – | 0.22757 |

Figure 4.1.3: Summary view of my proposed architectures as presented in CLEF 2022.

## 4.2 Additional experiments on IU X-RAY

In this section, we elaborate on several additional expepiments that we performed on IU X-RAY dataset [14] that constitutes one of the most popular publicly available biomedical datasets, containing both medical images and diagnostic reports, in order to investigate the consistency of our proposed architectures' performance in different settings including noticeable class imbalance. As mentioned in section 3.1.2, we took advantage of both IMPRESSION and FINDINGS fields where they exist, to generate the images' captions and the professional shall consider the remaining fields that are precisely described hereunder:

- `FINDINGS`: The visual characteristics of a body structure of function that may potentially have a clinical impact.
- `IMPRESSION`: The most remarkable findings as well as their clinical value. Might include a conclusion not followed by other sections and the images of the exam but we may ignore this issue for our current work.
- `COMPARISON`: Includes given information about the patient's previous illnesses. Might contain some information about the patient but **not a complete report** (e.g. extracts from previous exams).
- `INDICATION`: Contains given information regarding the medical reason subjected to examination (e.g. symptoms).

We found 104 reports with no associated image, 25 reports with empty Impression and Findings sections, 6 reports with no Impression section and 489 reports with no Findings section after collecting 7430 image-caption pairs to perform our training. Regarding concept detection, we took advantage of our backbone networks described in section 1.5 and added again a classification head, which we similarly fine-tuned on IU X-RAY. For caption generation, we reproduced the 1-NN Network as a baseline model. Recall that, if we denote by $\mathbf{e}(.)$ the output of the employed image encoder among those mentioned in section 1.5, 1-NN predicts $(\hat{x}, \hat{y}) = (\hat{x}, y^*)$ that satisfies $(x^*, y^*) = \arg\min_{\hat{x}} \cos(\mathbf{e}(\hat{x}), \mathbf{e}(x^*))$.

Although this heuristic is very simple, it produced rather high scores on IU-XRAY, which is further confirmed in the study [59], where it is also shown that it is capable of outperforming some much more elaborate approaches in clinical recall. This indicates already severe class imbalance.

Once again in that case, we have set in advance all the random seeds equal to $0$, the CUDNNs backends as deterministic and disabled the CUDNNs backends benchmark to ensure consistency of the aforementioned splits in consecutive runs for hyperparameter selection. This procedure has been applied during pre-processing for both subtasks re-ran using IU X-RAY.

### 4.2.1 Class imbalance on IU X-RAY: detection and handling

In this section, we are dealing with class imbalance issues using statistical significance tests, in order to identify to what extent this phenomenon occurs but also how the simple baselines are affected in terms of performance when up-sampling or down-sampling the majority class. As it has also been indicated in the above subsections, consecutive runs of the same algorithm as part of $5 \times 2$–fold cross validation scheme yield similar results and that can also be verified after we run statistical significance tests. The results are analyzed using T–tests but moreover, statistical significance tools are also being used to identify increase in performance of 1-NN in comparison to the more simplistic WinnerTakesAll (WTA) baseline.

Consequently, apart from the 1-NN network, an additional WinnerTakesAll baseline is involved and was also implemented for Diagnostic Captioning experiments, which uses the words frequency in the training captions and takes them in descending order, s.t. $f_i \geq f_{i+1}$, in order to generate the same caption for all instances of the test set. This was first introduced as the Frequency baseline [58] and thus commonly known as such within image captioning research community.

One maybe premature thought on investigating performance of the aforementioned fundamental algorithms or baselines is to perform several iterations to collect scores while up–sampling or down–sampling the majority class that in our case is the reports including no findings. The initial experiments aim to shed light exactly on investigating this issue, vaguely determined as whether the system is better at being a bad or a good doctor, taking advantage of the IMPRESSION and FINDINGS fields that are anyway those taken advantage of in this work but also by other researchers as well to perform Diagnostic Captioning on the X-RAYs.

In order to implement the up–sampling or down–sampling of the majority class, we adopted a naïve modelling of the class instances as reports containing the tokens *no findings*, *clear*, *normal* or some combination of them in either the IMPRESSION or the FINDINGS section, as well as in both, if they exist in the respective datapoints. We label as class $c_1$ the datapoints containing any image data (practically useless), as class $c_2$ the datapoints containing any textual information, class $c_3$ the datapoints containing non-null FINDINGS section, class $c_4$ the datapoints containing non-null IMPRESSION section, class $c_5$ the datapoints containing non-null both FINDINGS and IMPRESSION sections. Following this notation, the caption (target) generation procedure is explained hereunder:

**Procedure $P_1$: Preprocessing and caption generation**

```
Input:   an image $\vec{x} \in \mathcal{D}$
Output:  a pair $\vec{x}, \vec{y}$ of the image $\vec{x} \in \mathcal{D}$ and its associated caption $\vec{y} \in \mathcal{C}$

if $class(\vec{x}) = c_3$ set the caption $\vec{y} = fields.FINDINGS(\vec{x})$
if $class(\vec{x}) = c_4$ set the caption $\vec{y} = fields.IMPRESSION(\vec{x})$
if $class(\vec{x}) = c_5$ set the caption $\vec{y} = fields.IMPRESSION(\vec{x})$ + “ ” +
$fields.FINDINGS(\vec{x})$

return $\vec{x}, \vec{y}$
```

After generating the corresponding captions for the data in IU X-RAY, we divide the experiment process in four phases following each other, among which the positive phases relate to including only specific tokens $\hat{\mathcal{T}}$, either at least one or a combination of them as indicated in $\mathcal{T}$ whereas the negative phases relate to not including them, $\mathcal{T} = \cup_i\{t_i\} \in \hat{\mathcal{T}}$ such that $\hat{\mathcal{T}}$ = {no findings, clear, normal}. These tokens are mentioned earlier to characterize the majority class. In section 4.2.2, we are further expanding the experiment demonstrated in the previous subsection, using a larger set $\mathcal{T}^{(\text{freq})}$ with the most frequent tokens across all captions based on an extensive preprocessing pipeline concluding to a frequency analysis.

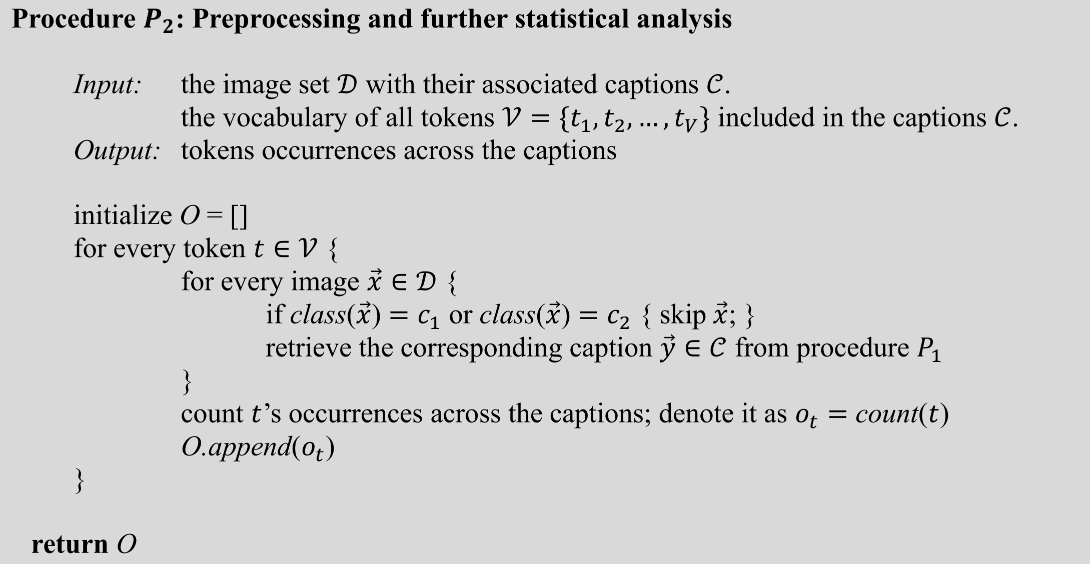
The outcome of this experiment is that the baselines perform better when expected to diagnose specific cases belonging to the majority class, by producing the respective targets, mostly including tokens in $\mathcal{T}$rather than when omitting all examples including either some or all tokens in $\mathcal{T}$ that indicate an X-RAY without clinical remarks or any medical reason subjected to examination. In other words, if we reduce the number of data belonging to the majority class (i.e. reports without clinical problems) the system performance deteriorates.

```
Algorithm $E_1$: Up–sampling/Down–sampling the negative class

Input:   the image set $\mathcal{D}$ with their associated captions $\mathcal{C}$.
         a set of tokens $\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$ to be removed from the captions among
         the train set, s.t. $\forall t \in \mathcal{T} \Rightarrow t \notin \mathcal{C}$
Output:  Precision, Recall, $F_1$ scores

define $\mathcal{T} = \{$“no findings”, “clean”, “normal”$\}$
for every image $\vec{x} \in \mathcal{D}$ {                      Phase $A^-$: any label included
    if $class(\vec{x}) = c_1$ or $class(\vec{x}) = c_2$ { skip $\vec{x}$; }
    retrieve the corresponding caption $\vec{y} \in \mathcal{C}$ from the above procedure
    if $\exists t \in \mathcal{T}$ s.t. $t \in \vec{y}$ { skip $\vec{x}$; }
}
train on the remaining pairs and generate Precision, Recall, $F_1$ scores

for every token $t \in \mathcal{T}$ {                           Phase $E^-$: every label included
    for every image $\vec{x} \in \mathcal{D}$ {
        if $class(\vec{x}) = c_1$ or $class(\vec{x}) = c_2$ { skip $\vec{x}$; }
        retrieve the corresponding caption $\vec{y} \in \mathcal{C}$ from the above procedure $(P_1)$
        if $t \in \vec{y}$ { skip $\vec{x}$; }
    }
    train on the remaining pairs and generate Precision, Recall, $F_1$ scores
}
for every image $\vec{x} \in \mathcal{D}$ {                      Phase $A^+$: any label not included
    if $class(\vec{x}) = c_1$ or $class(\vec{x}) = c_2$ { skip $\vec{x}$; }
    retrieve the corresponding caption $\vec{y} \in \mathcal{C}$ from the above procedure $(P_1)$
    if $\exists t \in \mathcal{T}$ s.t. $t \notin \vec{y}$ { skip $\vec{x}$; }
}
train on the remaining pairs and generate Precision, Recall, $F_1$ scores

for every token $t \in \mathcal{T}$ {                           Phase $E^+$: every label not included
    for every image $\vec{x} \in \mathcal{D}$ {
        if $class(\vec{x}) = c_1$ or $class(\vec{x}) = c_2$ { skip $\vec{x}$; }
        retrieve the corresponding caption $\vec{y} \in \mathcal{C}$ from the above procedure $(P_1)$
        if $t \notin \vec{y}$ { skip $\vec{x}$; }
    }
    train on the remaining pairs and generate Precision, Recall, $F_1$ scores
}
```

### 4.2.2 Extended frequency analysis of IU X-RAY captions

When introducing the problem of class imbalance, we split the IU X-RAY data into five distinct classes, according to their contained fields in the sense of them being non-null. One step further, among all the 3851 reports, 599 (15.6%) do not include a COMPARISON section, 86 (2.2%) do not include the INDICATION section, 31 (0.1%) do not include IMPRESSION, while 514 (13.3%) do not include FINDINGS. Among the reports that comprise both a FINDINGS section and at least one image (3337), only 2553 (76.5%) have a unique FINDINGS section, which means that the text of their FINDINGS is not the same in any other report. The FINDINGS section of the remaining $23.5\%$ reports is exactly the same in two or more other reports, and in these cases the reports describe mainly normal findings. In this setting reports related to illnesses, e.g. locations depicting an organ or tissue that suffered damage through injury or disease, are extremely limited.

For example, the most frequent FINDINGS text (found in $51$ reports) is “The heart is normal in size. The mediastinum is unremarkable. The lungs are clear.”. The 10 most frequent FINDINGS sections, all describing normal findings, occur in 344 reports in total, which is $10.3\%$ of all the reports and $43.9\%$ of the non-unique FINDINGS leading to the aforementioned class imbalance issue, but also serving as an alarming indicator of vocabulary shortage in the captions. In this context, we are expanding the experiment demonstrated in the previous subsection, using a larger set $\mathcal{T}^{(\text{freq})}$ with the most frequent tokens across all captions identified by the procedure explained below concluding to a standard frequency analysis.



**Procedure $P_2$: Preprocessing and further statistical analysis**

```
    Input:   the image set $\mathcal{D}$ with their associated captions $\mathcal{C}$.
             the vocabulary of all tokens $\mathcal{V} = \{t_1, t_2, \dots, t_V\}$ included in the captions $\mathcal{C}$.
    Output:  tokens occurrences across the captions

    initialize $O = []$
    for every token $t \in \mathcal{V}$ {
             for every image $\vec{x} \in \mathcal{D}$ {
                      if $class(\vec{x}) = c_1$ or $class(\vec{x}) = c_2$ { skip $\vec{x}$; }
                      retrieve the corresponding caption $\vec{y} \in \mathcal{C}$ from procedure $P_1$
             }
             count $t$'s occurrences across the captions; denote it as $o_t = count(t)$
             $O.append(o_t)$
    }

  return $O$
```

The questions we aim to answer through this process include whether the system is better (in terms of retrieval evaluation considering its generating captions) in specific subgroups of input data, and how does isolating those most frequent terms (i.e. such that $o_t > \epsilon$, where $\epsilon \in \mathcal{R}^+$ is a chosen frequency threshold) affect the WinnerTakesAll and 1-NN baselines performance. Evaluation includes the usual metrics and statistical significance tests mentioned in section 4.4.

**Algorithm $E_2$: Identifying groups of relevant captions and comparing**

```
Input:  the image set $\mathcal{D}$ with their associated captions $\mathcal{C}$.
        the vocabulary of all terms $\mathcal{V} = \{t_1, t_2, \dots, t_V\}$ included in the captions $\mathcal{C}$.
Output: Precision, Recall, F1 scores

    for every token $t \in \mathcal{V}$ {
        retrieve $t$'s occurrences across the captions $o_t$ from procedure $P_1$
            if $o_t$ is not sufficiently large number { skip $t$; }
            for every image $\vec{x} \in \mathcal{D}$ {
                if $class(\vec{x}) = c_1$ or $class(\vec{x}) = c_2$ { skip $\vec{x}$; }
                retrieve the corresponding caption $\vec{y} \in \mathcal{C}$ from procedure $P_2$
                if $t \notin \vec{y}$ { skip $\vec{x}$; }
            }
            train on the remaining pairs and generate Precision, Recall, F1 scores
        }
```

The tokens satisfying the criterion that $o_t$ is a sufficiently large number are $\mathcal{T}^{(\text{freq})}$= {no findings, clear, normal, acute, pleural, pneumothorax, effusion, heart, lungs, size, focal, pulmonary, cardiopulmonary, disease, limits, consolidation, abnormality, abnormalities, silhouette, mediastinal, cardiomediastinal, lung, airspace, stable, changes, chest, evidence, mild, spine, unremarkable, contour(s), thoracic, effusions, degenerative, atelectasis, calcified, upper, lobe, cardiac, opacity, opacities, vascularity, edema, intact, vasculature, vascularity, infiltrate, noted, bilateral, bilaterally, small, prior, pneumonia, interstitial}.

## 4.3 Statistical Significance Analysis outcomes for our ImageCLEFmedical 2022 contributions

In the following part, we present the outcomes from performing a variety of statistical significance tests to compare baselines to one another based on the assumption that they perform similarly; T–tests, F–tests, Wilcoxon sign-rank tests, Mann-Whitney U–tests, Cohen's effect size tests and Kolmogorov Smirnov tests, motivated in section 3.5, to many combinations of our implementations.

Our core aim from developing and training Deep Networks based on the convolutional image encoders and Generative Transformers [81] is to improve performance of simple baselines by an extent that is statistically significant on a predefined level $5 \times 10^{-2}$. This goal is satisfied according to all our comparisons for caption generation, apart from the case where we compare the 1-NN baseline with the hybrid model including summarization, whereas we observe that although RAG-token based systems perform better in the validation and held-out development sets, they fail to generalize well in the unknown test set, while abstractive summarization performs surprisingly better although it lacks a non-parametric memory. In contrast, the results are controversial for concept detection, in which case we elaborate on our observations.

We start by performing both one-tailed and two-tailed statistical significance T-tests to evaluate similarity of the $F_1$ scores among our experiments demonstrated in Table 7.1 and the submitted results presented in section 6.1.1 to compare classification baselines to one another and follow the same steps for the caption generation methods proposed in section 6.1.2. The tests' results are presented in chapter 7 (Appendix A), marked in red in cases where we reject the null hypothesis $H_0$ and green when we fail to do so, while regarding the effect size, we use orange when it is small, blue when it is medium and purple when it is large in order to navigate the reader.

## 4.3.1 Significance tests for all experiments outcomes

To begin with, we compare the different backbone network architectures. We observe that DenseNet is the best performing image encoder for concept detection, as it is also indicated by the validation $F_1$ scores, while the performance difference is statistically significant compared to AlexNet and ResNet image encoders but we cannot reject the null hypothesis when comparing to VGG. We also observe a high variance in the results related to DenseNet compared to AlexNet and ResNet image encoders; other than that Wilcoxon signed-rank tests and Mann-Whitney U tests agree that the performance difference is statistically significant as we illustrate below.

Table 4.3.1: P-Values and Cohen's d for different types of statistical tests between different backbone architectures.

| **Statistical test** | AlexNet / DenseNet | AlexNet / ResNet | AlexNet / VGG | DenseNet / ResNet | DenseNet / VGG | ResNet / VGG |
|---|---|---|---|---|---|---|
| One tailed *T*-test (default) | 0,03026 | 0,16456 | 0,05327 | 0,00103 | 0,09123 | 0,00425 |
| One tailed *T*-test, equal variance | 0,02469 | 0,10878 | 0,03269 | 0,0006 | 0,38864 | 0,00082 |
| One tailed *T*-test, unequal variance | 0,02111 | 0,10734 | 0,0048 | 0,00025 | 0,20551 | 4,13E-05 |
| Two tailed *T*-test (default) | 0,06053 | 0,32911 | 0,10655 | 0,00206 | 0,18246 | 0,0085 |
| Two tailed *T*-test, equal variance | 0,04938 | 0,21757 | 0,06537 | 0,0012 | 0,77727 | 0,00164 |
| Two tailed *T*-test, unequal variance | 0,04221 | 0,21469 | 0,0096 | 0,0005 | 0,41102 | 8,25E-05 |
| *F*-test (for stat. variance analysis) | 0,62866 | 0,56602 | 5,71E-08 | 0,25496 | 1,47E-08 | 1,97E-07 |
| One tailed Mann-Whitney U-test | 2,95E-06 | 0,00142 | 0,00107 | 7,84E-09 | 0,03793 | 5,90E-06 |
| Two tailed Mann-Whitney U-test | 5,90E-06 | 0,00284 | 0,00214 | 1,57E-08 | 0,07585 | 1,18E-05 |
| Cohen's d - Baseline Effect Size test | -0,4726 | 0,38705 | -0,7295 | 0,82278 | -0,0946 | -1,3492 |
| One tailed Wilcoxon signed-rank test | 2,90E-06 | 0,00136 | 0,001 | 7,68E-09 | 0,03746 | 5,43E-06 |
| Two tailed Wilcoxon signed-rank test | 5,81E-06 | 0,00273 | 0,002 | 1,54E-08 | 0,07493 | 1,09E-05 |
| Kolmogorov-Smirnov test | 5,08E-07 | 0,00018 | 0,01701 | 6,83E-12 | 0,00332 | 5,73E-07 |

In addition, we compare architectures where we use pre-trained backbone networks on ImageNet classification dataset with baselines, while we include the same image encoders combined with a heuristic approach based on 1-NN. In that case, we again observe a high variance in the results related to DenseNet and ResNet image encoders, as well as that the fine-tuned classification heads' performance, which are initialized using Glorot formula as explained in previous parts, is better and the $F_1$ difference is statistically significant for any network architecture.

Moreover, we vary the training extent and compare models where we fully finetune the CNNs end-to end using the same objective and cyclical learning rates with architectures where we load pre-trained backbones on ImageNet classification dataset and systems where we include the same image encoders combined with a heuristic approach based on 1-NN. Again, we notice a high variance and observe that the finetuned classification heads' performance, which are initialized using Glorot formula as explained previously, yields higher scores than the fully finetuned vision encoders and both are better than the 1-NN baseline The $F_1$ score differences in both cases are statistically significant for any network architecture (either DenseNet121 or DenseNet161).

Last but not least, we also compare architectures where we use pre-trained backbone networks on ImageNet classification dataset with their ensembles, attempting to take advantage of the "Wisdom of the crowd" [76]. In that scenario we observe that the fine-tuned classification heads' performance, which are initialized using Glorot formula as explained in previous parts, perform better when we include the same image encoders and the F1 difference is statistically significant but we cannot reject the null hypothesis when comparing to the top-10 best performing DenseNets ensemble. In the latter case, the effect size is small, while we take into consideration only DenseNet weak learners for a fair performance comparison; yet we fail to reject our null hypothesis $H_0$. All the respective results are provided in chapters 6 and 7 (Appendices).

### 4.3.2 Significance tests for submitted baselines

Furthermore, we perform the same range of statistical significance tests to compare submissions to ImageCLEF to one another; namely T–tests, F–tests, Wilcoxon sign-rank tests, Mann-Whitney U–tests, Cohen's effect size tests and Kolmogorov Smirnov tests as we describe in section 3.5. Although limited data is available for this purpose, this gives us the opportunity to take account of the Manual F1 score in the statistical significance analysis, which was provided by human annotators for concept detection, as well as SPICE and BERTscore in the case of caption generation, as it is demonstrated again in chapter 7 (Appendix B).

To begin with, we start by comparing submissions where we use pre-trained backbone networks on ImageNet classification dataset with their ensembles, attempting to take advantage of the "Wisdom of the crowd" [76]. In that scenario we observe that the fine-tuned classification heads' performance, which are initialized using Glorot formula as explained in previous parts, perform better when we include the same image encoders and the F1 difference is statistically significant according to most tests. There variance is high in the manual F1 score, however Wilcoxon signed-rank tests and Mann-Whitney U tests agree the performance difference is statistically significant, therefore the fine-tuned classification heads perform better than other baselines and the F1 difference is statistically significant.

What is more, we compare submissions where we use pre-trained backbone networks on ImageNet classification dataset with fully finetuned CNNs end-to-end using the same objective and cyclical learning rates with other architectures where we use pre-trained backbone networks on ImageNet classification dataset and baselines where we include the same image encoders combined with a heuristic approach based on 1-NN. According to the T-tests, we need to make the assumption of equal variance in order to reject the null hypothesis and although the F-test indicates a low variance in the F1 scores we fail to reject the null hypothesis according to the Wilcoxon signed-rank tests and Mann-Whitney U tests additionally performed.

In addition, we try different training regimes and compare our submissions where we use fully finetuned CNNs end-to-end using the same objective and cyclical learning rates with architectures where we use pre-trained backbone networks on ImageNet classification dataset and baselines where we include the same encoders combined with a heuristic approach based on 1-NN to ensemble networks. Again we need to make the prior assumption of equal variance in order to reject the null hypothesis and yet we fail to do so considering the Wilcoxon signed-rank and Mann-Whitney U tests.

This section provides an overview of indicative outcomes; consequently, if you aim to obtain a thorough understanding of our proposed methods and get additional details on their performance statistics, we strongly recommend that you also read the text in chapter 6 and study the tables in chapter 7.

## 4.4 Statistical Significance Analysis outcomes for our additional experiments on IU X-RAY

Furthermore, we perform a subset of the aforementioned statistical significance tests to compare WinnerTakesAll and 1-NN baselines in IU X-RAY to one another; namely T–tests, F–tests, Mann-Whitney U–tests. We additionally apply the same range of tests in the context of the experiment described in section 4.2.2 related to class imbalance detection and investigation of its extent.

### 4.4.1 Baselines comparison and performance ordering

In this section we take advantage of the multiple runs performed in the context of $5 \times 2$-fold cross validation in section 6.2.1 to iterate over the same procedure and perform both one-tailed and two-tailed statistical significance T-tests to evaluate similarity of the respective results again and based on the assumption (the null hypothesis $H_0$) that they perform similarly, compare their performance using the usual metrics described in section 3.4 with their aforementioned limitations. Although the requirements for the T–test for two paired samples are here satisfied, the Wilcoxon Signed-Rank Test for Paired Samples non-parametric test will also be used or in particular some variation called Mann-Whitney U–test. Performing a Signed-Rank Test, as well as a variance test are both proved to support the T–test conclusions that there is a statistically significant difference in the networks in performance.

Table 4.4.1: P-Values for different types of statistical tests between WinnerTakesAll and 1-NN baselines for 90% train-10% validation split when performing 5 × 2–fold cross validation.

| **Statistical test** | BLEU 1 | BLEU 2 | BLEU 3 | BLEU 4 | METEOR | ROUGE | CIDER |
|---|---|---|---|---|---|---|---|
| One tailed *T*-test (default) | 2.45753e-08 | 1.19898e-07 | 3.3071e-08 | 9.64336e-09 | 6.23903e-09 | 2.37431e-05 | 0.00 |
| One tailed *T*-test, equal variance | 4.12503e-14 | 2.83594e-12 | 4.81584e-14 | 1.2627e-14 | 3.9743e-12 | 3.33685e-07 | 7.91361e-06 |
| One tailed *T*-test, unequal variance | 7.91481e-13 | 1.03159e-09 | 8.56087e-12 | 9.64336e-09 | 4.02936e-12 | 4.94759e-06 | 1.90541e-05 |
| Two tailed *T*-test (default) | 4.91506e-08 | 2.39797e-07 | 6.61421e-08 | 1.92867e-08 | 1.24781e-08 | 4.74862e-05 | 0.00 |
| Two tailed *T*-test, equal variance | 8.25007e-14 | 5.67188e-12 | 9.63169e-14 | 2.52541e-14 | 7.94861e-12 | 6.6737e-07 | 1.58272e-05 |
| Two tailed *T*-test, unequal variance | 1.58296e-12 | 2.06318e-09 | 1.71217e-11 | 1.92867e-08 | 8.05872e-12 | 9.89517e-06 | 3.81081e-05 |
| *F*-test (variance analysis) | 0.41 | 0.13 | 0.24 | 0.00 | 0.96 | 0.13 | 0.36 |
| Mann Whitney *U*-test | 4.92e-03 | 5.00e-03 | 4.55e-03 | 2.78e-03 | 5.07e-03 | 4.92e-03 | 5.05e-03 |

Of course, what is concluded after performing the statistical significance tests, while making the hypothesis $H_0$ is that we get the expected order of performance for the majority of our evaluation metrics, which is $\mathcal{P}$(1-NN) > $\mathcal{P}$(WinnerTakesAll), supposing that $\mathcal{P}$ is a function considering all the metrics to measure overall performance based on the employed metrics.

### 4.4.2 Statistical significance tests to verify class imbalance

In this section, we iterate experiments with WinnerTakesAll and 1-NN baselines on IU X-RAY using different ratios of train and test data (i.e. 90-10, 80-20, 95-5), perform both one-tailed and two-tailed statistical significance T-tests to evaluate similarity of the respective results and based on the assumption (i.e. our usual null hypothesis $H_0$) that they perform similarly. However, both baselines perform better when applied to the majority class, which indicates the class imbalance characterizing the dataset and highlights its extent. The experimental setting and the definition of the majority class tokens $\mathcal{T}^{(\text{freq})}$ is according to the methodology described in section 4.2.2 for detecting class imbalance in IU X-RAY.

Table 4.4.2: *P*-Values for different types of *T*-test between WinnerTakesAll and 1-NN baselines for 90-10 split when necessitating certain tokens belonging to data of the majority class; $\mathcal{T}^{(\text{freq})}$.

| ***T*-test type** | BLEU 1 | BLEU 2 | BLEU 3 | BLEU 4 | METEOR | ROUGE | CIDER |
|---|---|---|---|---|---|---|---|
| One tailed *T*-test | 2.5036e-42 | 9.20886e-29 | 7.74274e-30 | 4.36437e-21 | 3.53333e-32 | 5.24133e-23 | 0.48 |
| Same, equal var. | 8.69483e-25 | 5.2273e-25 | 2.83401e-28 | 2.3178e-21 | 2.13204e-11 | 3.92138e-15 | 0.48 |
| Same, unequal var. | 1.27279e-24 | 8.13582e-24 | 1.92232e-24 | 1.43544e-17 | 2.67555e-11 | 3.25331e-14 | 0.48 |
| Two tailed *T*-test | 5.0072e-42 | 1.84177e-28 | 1.54855e-29 | 8.72874e-21 | 7.06666e-32 | 1.04827e-22 | 0.95 |
| Same, equal var. | 1.73897e-24 | 1.04546e-24 | 5.66802e-28 | 4.63561e-21 | 4.26407e-11 | 7.84277e-15 | 0.96 |
| Same, unequal var. | 2.54559e-24 | 1.62716e-23 | 3.84465e-24 | 2.87088e-17 | 5.3511e-11 | 6.50662e-14 | 0.96 |

Table 4.4.3: *P*-Values for different types of *T*-test between WinnerTakesAll and 1-NN baselines for 80-20 split when necessitating certain tokens belonging to data of the majority class; $\mathcal{T}^{(\text{freq})}$.

| ***T*-test type** | BLEU 1 | BLEU 2 | BLEU 3 | BLEU 4 | METEOR | ROUGE | CIDER |
|---|---|---|---|---|---|---|---|
| One tailed *T*-test | 3.60922e-29 | 6.16074e-22 | 4.23632e-22 | 1.94061e-16 | 5.56464e-15 | 3.85457e-16 | 0.11 |
| Same, equal var. | 1.33675e-24 | 6.24051e-28 | 3.90119e-29 | 1.56676e-20 | 4.76593e-12 | 5.57635e-15 | 0.11 |
| Same, unequal var. | 7.91575e-24 | 2.47966e-24 | 6.67909e-23 | 1.92467e-16 | 1.947e-11 | 1.62897e-13 | 0.11 |
| Two tailed *T*-test | 7.21844e-29 | 1.23215e-21 | 8.47264e-22 | 3.88122e-16 | 1.11293e-14 | 7.70913e-16 | 0.23 |
| Same, equal var. | 2.6735e-24 | 1.2481e-27 | 7.80237e-29 | 3.13351e-20 | 9.53186e-12 | 1.11527e-14 | 0.23 |
| Same, unequal var. | 1.58315e-23 | 4.95932e-24 | 1.33582e-22 | 3.84933e-16 | 3.894e-11 | 3.25794e-13 | 0.23 |

Table 4.4.4: *P*-Values for different types of *T*-test between WinnerTakesAll and 1-NN baselines for 95-5 split when necessitating certain tokens belonging to data of the majority class; $\mathcal{T}^{(\text{freq})}$.

| ***T*-test type** | BLEU 1 | BLEU 2 | BLEU 3 | BLEU 4 | METEOR | ROUGE | CIDER |
|---|---|---|---|---|---|---|---|
| One tailed *T*-test | 9.90699e-40 | 3.55036e-27 | 6.23996e-30 | 1.07275e-26 | 1.35728e-31 | 1.93063e-21 | 0.05 |
| Same, equal var. | 1.14003e-25 | 2.98285e-34 | 4.75008e-44 | 1.61474e-37 | 3.59908e-16 | 3.70078e-17 | 0.04 |
| Same, unequal var. | 1.14414e-25 | 5.10128e-31 | 2.88486e-33 | 9.35262e-27 | 5.02566e-16 | 5.88941e-16 | 0.04 |
| Two tailed *T*-test | 1.9814e-39 | 7.10073e-27 | 1.24799e-29 | 2.14549e-26 | 2.71457e-31 | 3.86126e-21 | 0.10 |
| Same, equal var. | 2.28006e-25 | 5.96571e-34 | 9.50016e-44 | 3.22948e-37 | 7.19817e-16 | 7.40155e-17 | 0.08 |
| Same, unequal var. | 2.28827e-25 | 1.02026e-30 | 5.76972e-33 | 1.87052e-26 | 1.00513e-15 | 1.17788e-15 | 0.08 |

Table 4.4.5: *P*-Values for different types of tests between WinnerTakesAll and 1-NN baselines for various splits when necessitating certain tokens belonging to data of the majority class; $\mathcal{T}^{(\text{freq})}$.

| **Statistical test** | BLEU 1 | BLEU 2 | BLEU 3 | BLEU 4 | METEOR | ROUGE | CIDER |
|---|---|---|---|---|---|---|---|
| Variance *F*-test (90-10) | 0.28 | 2.27099e-03 | 8.29678e-09 | 8.03804e-22 | 6.67432e-02 | 2.97278e-06 | 6.71613e-06 |
| Variance *F*-test (80-20) | 0.02 | 2.55703e-08 | 2.10604e-20 | 1.12296e-48 | 1.29187e-06 | 1.36244e-11 | 1.74146e-45 |
| Variance *F*-test (95-5) | 0.92 | 5.60704e-05 | 5.8265e-16 | 8.55346e-49 | 0.14 | 2.03518e-06 | 8.03894e-04 |
| Mann-Whitney *U*-test (90-10) | 0.00 | 0.00 | 0.00 | 0.00 | 1.25344e-12 | 5.79536e-14 | 2.47732e-03 |
| Mann-Whitney *U*-test (80-20) | 0.00 | 0.00 | 0.00 | 0.00 | 3.04201e-13 | 1.08802e-14 | 0.63 |
| Mann-Whitney *U*-test (95-5) | 0.00 | 0.00 | 0.00 | 0.00 | 1.27764e-12 | 1.22125e-14 | 7.098501e-03 |

# Chapter 5

# Discussion and conclusions

In this work, we developed CNN-based vision encoders and trained them for concepts assignment or combined them with heuristic-based methods such as the 1-NN baseline for either tags prediction or caption generation [50], while in summary we principally employed three training regimes; finetuning the CNNs' classification heads in the case of concept detection and keeping the rest of each backbone network pretrained on ImageNet classification dataset [70], fully fine-tuning the image encoders or just using pretrained architectures combined with heuristic approaches such as the 1-NN baseline as structural units.

In the case of concept detection, the first approach is the best performing and results in higher $F_1$ scores, where we set as training objective of the classification heads the negative $F_1$ scores and even outperforms network ensembles, where we attempted to take advantage of the "Wisdom of the crowd" [76] for the fully fine-tuned models using a majority voting policy, since other approaches such as the concepts' union or their intersection performed poorly. Also neither the 1-NN baseline nor its generalization to $k$-NN did result in competitive scores, although using weights depending on the cosine similarities of the retrieved images for the concept assignments seems a probable promising direction for bringing retrieval-based approaches at the forefront of medical tagging together with Deep Networks [7].

Regarding caption generation baselines, we took a lot advantage of our aforementioned generalization of 1-NN baseline, where we take into account multiple visually similar images instead of just the closest one, which although is really simple performs rather well if combined with abstractive summarization algorithms, as highlighted in section

4.1.2 as well as the study in [59], where it performs surprisingly well for the Indiana University chest X-ray Collection [14] (IU X-RAY) only by itself. We also attempted to combine the 1-NN baseline with Retrieval Augmented Generation (RAG) [45], in order to combine the success of parametric sequence-to-sequence models with the strengths of Dense Passage Retriever (DPR) [34] that designates modern Information Retrieval [45]. Essentially, apart from the parametric memory in the weights of the generative transformer, we tried adding an additional non-parametric memory in the form of a FAISS index built on $40\%$ of PubMed 2022, the performance obtained by this model however was lower at test time.

Last but not least, the surprising performance of the 1-NN baseline in IU X-RAY was verified by our own experiments and in our opinion is due to the severe class imbalance, which we detected by performing a frequency analysis. Furthermore, we apply a wide range of statistical tests to quantify the statistical significance of the performance gap observed between different baselines based on the assumption (null hypothesis $H_0$) that they perform similarly; namely T–tests, F–tests, Wilcoxon sign-rank tests, Mann-Whitney U–tests, Cohen's effect size tests and Kolmogorov Smirnov tests to various combinations of our implementations. Our baselines based on the convolutional image encoders and Generative Transformers [81] performed competitively in ImageCLEF Medical 2022 and the performance gap is statistically significant on a predefined level $5 \times 10^{-2}$ when comparing to simple baselines.

## 5.1 Future Work

Future work could focus more on the use of task-specific models for summarization, such as Bio-BERT [43] or BlueBERT [61], additional fine-tuning for the amount of neighbours $k$ and the summary maximum length $n$ in section 4.1.2 and consideration of potential associations between the two subtasks during 1-NN baseline [50] extension. Moreover, although higher quantitative accuracy is most often better, there are also categorical differences of the DC methods, which relate to their qualitative evaluation and indicate their practical usefulness. It is an open question whether and how we may obtain practical information about the quality of the generated captions. In section 5.3, we provide indicative generations of our best-performing baselines based on Pegasus abstractive summarizer [91], however qualitative evaluation by experts such as medical doctors and radiologists remains as future work.

## 5.2 Best submissions in ImageCLEF medical 2022

In order to qualitatively compare our submissions to the best of labs in ImageCLEF medical 2022, we present the best performing systems architectures, as well as the key differences between them and our proposed baselines. Furthermore, it is crucial thus important to apprehend that the key underlying architectures are similar in the case of concept detection, with several differences in the choice of loss function and backbone architecture being responsible for the performance difference, while a simple LM based on a classification framework achieved the highest scores in caption generation, similar to our tagging systems but based on the captions vocabulary $\mathcal{V}$.

Starting with the concept prediction subtask, the best performing group also employed ensembles of CNN image encoders, pre-trained on ImageNet [70] and then finetuned in the X-RAY images, as one of their principal components to codify the images into dense representations; to seek for diversity and in order to exploit the "Wisdom of the crowd" [76] for the fine-tuned models. However, their checkpoints were obtained from `keras` library rather than `torchvision` and their best performing backbone network is EfficientNetB0, which was not included among the architectures that we tried in our experiments illustrated in Figure 1.5.1 and it is followed by Generalized-Mean global pooling [63], as well as controlled by tunable thresholds in the sigmoid activations of the different classes, boosting their models' performance [8].

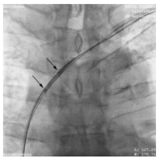
Furthermore, they successfully developed more complex voting schemes based on the union and the intersection of the weak learners' predicted concept sets, rather than a simple majority voting [8] and multiplied the negative $F_1$ score —that we also used— by the Binary Cross Entropy in the loss function further improving their performance [7]. What is more, they also experimented with other loss functions such as the Focal Loss, Assymetric Loss, soft $F_1$ and Sharphness Aware Minimization, which however did not bring them any additional benefits with respect to the $F_1$ score. Last but not least, similarly to our heuristic approaches extending the 1-NN baseline [50], they proposed an interesting weighting mechanism depending on the cosine similarities of the retrieved images for the concept assignments [7].

Regarding the caption generation subtask, the best performing approach was a binary classifier similar to our concept detection baselines based on the captions' tokens $t \in \mathcal{V}$, which however does not generate consistent outputs but rather extracts keywords that are relevant to the respective medical images [24].

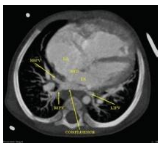
## 5.3 Indicative generations of our systems

Last but not least, in order to provide the possibility of minimal qualitative evaluation of our proposed baselines a posteriori to their submission in ImageCLEF Medical 2022, as well as to the publication of this thesis, we provide indicative captions as predicted by our best performing system, together with the ground truth captions and the respective radiological images. Although there are cases in which the system predicts exactly the expected captions, as in the examples shown below, there also are many cases where it may provide additional information that is not necessarily accurate, either by adding additional sentences to the caption of the visually most similar image due of the $k$-NN summarization or by incorporating it in the caption. In addition, the predicted captions may also differ a lot to the ground truth.

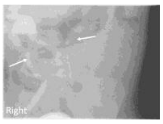
**Exactly or nearly identical reports**



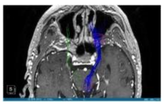
*ImageCLEFmedCaption_2022_train_003212.jpg*
**Ground truth:** balloon dilatation of left subclavian proximal stenosis
**Prediction:** balloon dilatation of left subclavian proximal stenosis



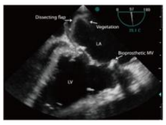
*ImageCLEFmedCaption_2022_train_081401.jpg*
**Ground truth:** computerized tomographic image show drainage of leave inferior pulmonary vein into confluence large atrial septal defect and confluence behind leave atrium
**Prediction:** computerized tomographic image show drainage of leave inferior pulmonary vein into confluence large atrial septal defect and confluence behind leave atrium



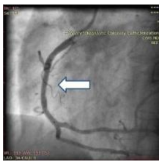
*ImageCLEFmedCaption_2022_train_068453.jpg*
**Ground truth:** case a electrocardiogram with inferolateral early repolarization pattern with jpoint elevation and qrs slur after hypothermia treatment red arrow
**Prediction:** case a electrocardiogram with inferolateral early repolarization pattern with jpoint elevation and qrs slur after hypothermia treatment red arrow



*ImageCLEFmedCaption_2022_train_043443.jpg*
**Ground truth:** mri axial view dbs blue right mfb green leave mfb
**Prediction:** mri axial view dbs blue right mfb green leave mfb

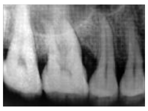


*ImageCLEFmedCaption_2022_train_026749.jpg*
**Ground truth:** transesophageal echocardiography in mid esophageal two chamber view degree demonstrate a linear dissection flap originate from posterior side of leave atrial wall la leave atrium lv leave ventricle mv mitral valve
**Prediction:** transesophageal echocardiography in mid esophageal two chamber view degree demonstrate a linear dissection flap originate from posterior side of leave atrial wall la leave atrium lv leave ventricle mv mitral valve

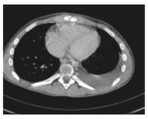


*ImageCLEFmedCaption_2022_train_065779.jpg*
**Ground truth:** male with acsinferior wall mi post thrombolysis and post mi angina coronary angiographyleave anterior oblique angiographic view of rca show end result post stenting arrow
**Prediction:** male with acsinferior wall mi post thrombolysis and post mi angina coronary angiograph show leave anterior oblique fluoroscopic view of rca during stent deployment arrow

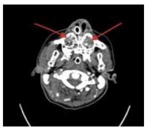
### Includes additional information



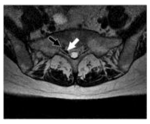
*ImageCLEFmedCaption_2022_train_021072.jpg*
**Ground truth:** radiographic osseous defect fill after month
**Prediction:** radiographic osseous defect fill after month We report a case of periapical radiograph defect in a



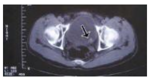
*ImageCLEFmedCaption_2022_train_047779.jpg*
**Ground truth:** ct of thorax with iv contrast post surgery
ct of thorax with iv contrast post surgery Compute tomography scan of the abdomen show extensive pneumoperitone



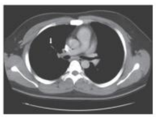
*ImageCLEFmedCaption_2022_train_046418.jpg*
**Ground truth:** the angioct axial view of the maxillary lesion
**Prediction:** the angioct axial view of the maxillary lesion Computed tomography (CT) and computed tomography (CT) images



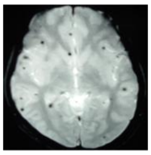
*ImageCLEFmedCaption_2022_train_036103.jpg*
**Ground truth:** mri mri show a low signal lesion identify as an air bubble white arrow adjacent to the right root black arrow
**Prediction:** mri mri show a low signal lesion identify as an air bubble white arrow adjacent to the right root black arrow Magnetic resonance imaging mri scan of the left ovary show a large



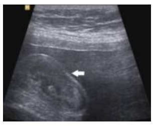
*ImageCLEFmedCaption_2022_train_052391.jpg*
**Ground truth:** computed tomography scan of the recurrent lesion with a size of as indicate by the arrow perivesical invasion be not see
**Prediction:** computed tomography scan of the recurrent lesion with a size of as indicate by the arrow perivesical invasion be not see Computed tomography scan of the primary lesion with a size of as



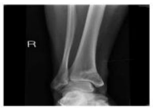
*ImageCLEFmedCaption_2022_train_061846.jpg*
**Ground truth:** chest ct scan
**Prediction:** case chest compute tomography show sized mediastinal cystic mass which be locate on the posterior to the low esophagus it be suspect to be an esophageal duplication cyst or a bronchogenic cyst Computed tomography (CT) scan of the abdomen and chest show



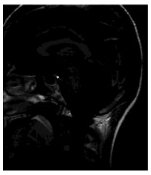
*ImageCLEFmedCaption_2022_train_040539.jpg*
**Ground truth:** mri brain gre show multiple hypointense lesion diagnose as cerebral cavernoma
**Prediction:** mri brain gre show multiple hypointense lesion diagnose as cerebral cavernoma Compute tomography scan with intravenous contrast show a large heterogeneous mass arise from



*ImageCLEFmedCaption_2022_train_030886.jpg*
**Ground truth:** longitudinal sonography of foetal scrotum show concave surface arrow
**Prediction:** longitudinal sonography of foetal scrotum show an alter contour of the scrotum and movement of intrascrotal echo the surface of the scrotum be less convex arrow and concave arrow in figure compare with figure where the surface be almost flat Here are some of the findings from a series of abdominal ultrasounds

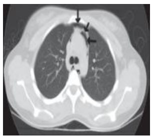


*ImageCLEFmedCaption_2022_train_067460.jpg*
**Ground truth:** ap xray of ankle show cystic lesion in distal tibia
**Prediction:** lateral view xray of the ankle show well define cystic lesion in distal tibia radiographs of the ankle, knee, and tibia have been used to



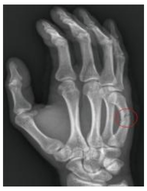
*ImageCLEFmedCaption_2022_train_043601.jpg*
**Ground truth:** enhanced magnetic resonance image mri of the sagittal section of patient show the narrowing of the nodular lesion below the hypothalamus region follow treatment
**Prediction:** enhanced magnetic resonance image mri of the sagittal section of patient show the narrowing of the pituitary stalk follow treatment the posterior pituitary continue to lack the characteristic high signal Magnetic resonance imaging scan of the brain of a man with acl

### Token repetition

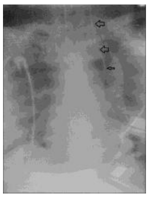


*ImageCLEFmedCaption_2022_train_054277.jpg*
**Ground truth:** rounded lucencie see within medastinal pleura surround the medastinal structure indicate by arrow
**Prediction:** rounded lucencie see within medastinal pleura surround the medastinal structure indicate by arrow Computed tomography scan Computed tomography scan Computed tomography scan Compute

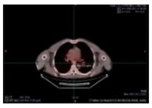
### Small errors



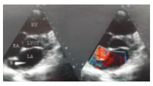
*ImageCLEFmedCaption_2022_train_009712.jpg*
**Ground truth:** preoperative xray of the first fracture event the xray show the fracture of the fifth metacarpal bone shaft of the left hand red circle
**Prediction:** preoperative xray of the second fracture event the xray show refracture and holedefect of the fifth metacarpal bone shaft of the left hand red circle Radiograph of the right hand show multiple expansile osteolytic



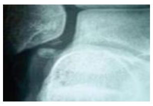
*ImageCLEFmedCaption_2022_train_022459.jpg*
**Ground truth:** chest radiograph anteriorposterior view of a female patient note the left paramedian location of the central venous catheter beside the aortic knuckle chest radiograph anteriorposterior view of a male patient note the left paramedian
**Prediction:** location of the central venous catheter beside the aortic knuckle



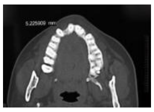
*ImageCLEFmedCaption_2022_train_075657.jpg*
**Ground truth:** the initial followup petct scan month after hospital discharge show fdg avid area in the subcarinal suv tissue represent pericardial involvement of kaposi sarcoma
**Prediction:** the initial followup petct scan month after hospital discharge show fdg avid area in the right paratracheal suv of tissue represent pericardial involvement of kaposi sarcoma Computed tomography ct scan image of the thorax shows a



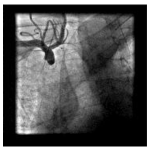
*ImageCLEFmedCaption_2022_train_072225.jpg*
**Ground truth:** parasternal short axis view show baffle drain into the leave atrium arrow bbaffle raright atrium rvright ventricle laleave atrium lvotleave ventricular outflow tract
**Prediction:** parasternal short axis view show baffle just below the level where it drain into the leave atrium bbaffle raright atrium rvright ventricle laleave atrium lvotleave ventricular outflow tract



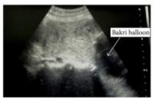
*ImageCLEFmedCaption_2022_train_021758.jpg*
**Ground truth:** radiograph of anteroposterior right fibular malleolus
**Prediction:** radiograph of anteroposterior medial right malleolus



*ImageCLEFmedCaption_2022_train_044723.jpg*
**Ground truth:** computed tomographic scan show the mesiodistal width of the available bone at the edentulous site
**Prediction:** computed tomographic scan show the labiopalatal width of the available bone at the edentulous site Here is a report of a successful total omentectomy for a

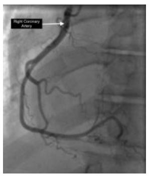


*ImageCLEFmedCaption_2022_train_003898.jpg*
**Ground truth:** angiogram of right subclavian artery occlusion of brachiocephalic artery
**Prediction:** angiogram of left subclavian artery occlusion of left subclavian artery above aortal outlet

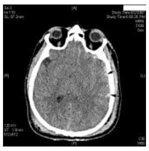


*ImageCLEFmedCaption_2022_train_030117.jpg*
**Ground truth:** the uterine fundus be push up with a balloon contain of physiological saline the shape of the uterine fundus be irregular and indistinct and complete reduction be not achieve
**Prediction:** the uterine fundus be again push up with a balloon contain of physiological saline the uterus be completely reduce and the hemorrhage be stop Ultrasonography of the abdomen using a detailed transthoracic

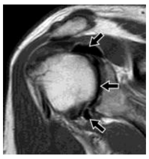
### Other cases of errors



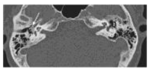
*ImageCLEFmedCaption_2022_valid_087667.jpg*
**Ground truth:** coronary angiogram of the right coronary system in left anterior oblique position show a patent right coronary artery
**Prediction:** coronary angiography show the anomalous origin of the right coronary artery from the left sinus of valsalva A coronary angiogram shows a stenosis of the proximal anterior leave



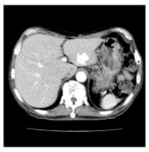
*ImageCLEFmedCaption_2022_train_045328.jpg*
**Ground truth:** computed tomography scan at the level of the orbit punctiform contusion involve the leave temporal and frontal lobe with effacement of the leave occipital horn be demonstrate arrow a anterior p posterior l leave r right
**Prediction:** computed tomography scan at the level of the midbrain multiple contusion involve the leave temporal lobe be evident arrow a anterior p posterior l leave r right The following is a description of some of the brain scans performed on



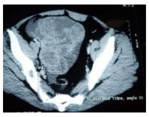
*ImageCLEFmedCaption_2022_train_074981.jpg*
**Ground truth:** se sequence trtethick mm on oblique coronal plane after adminbetration of dimeglumine salt of gadopentetate acid dilute in saline solution at moll concentration artefact be detectable characterize by contrast signal absence black arrow with no evidence of contrast agent in periarticular soft tbesues
**Prediction:** se sequence trtethick mm on oblique coronal plane after adminbetration of dimeglumine salt of gadopentetate acid dilute in saline solution at mmoll concentration the expect contrast signal hyperintensity be detectable black arrow capsular wall fibrous thickening at axillary pouch white arrow and supraspinatus tendon partial tear on articular side point white arrow be evident Magnetic resonance imaging mri scan of the patients right ankle show rupture



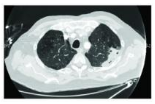
*ImageCLEFmedCaption_2022_train_077325.jpg*
**Ground truth:** an axial computed tomographic image show an osseous lesion in the right middle ear space arrow
**Prediction:** axial image of the right temporal bone show soft tissue mass lesion in the jugular fossa with irregular destruction of the jugular fossa white arrow jugular spine adjacent bone and extension of the mass into the middle ear cavity black arrow glomus jugulotympanicum Computed tomography scan of the head coronal view coronal view coronal view



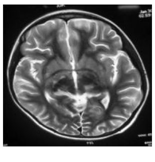
*ImageCLEFmedCaption_2022_train_046254.jpg*
**Ground truth:** enhance computed tomography ct of the abdomen the tumor be detect in the spiegel of the liver on enhance ct of the abdomen arrow
**Prediction:** contrast enhanced axial ct of the abdomen in a male with an inguinal trauma bowel wall thickening and enhancement Computed tomography ct scan take after admbesion the



*ImageCLEFmedCaption_2022_train_053378.jpg*
**Ground truth:** ct scan of the pelvis heterogeneous mass of the right ovary enhance after contrast material injection
**Prediction:** ct scan with oral and rectal contrast r show a pelvic mass with multiple heterogeneously enhance soft tissue masse with central necrotic area within pelvis not separately see from the ovary est A contrastenhanced computed tomography (ct) scan of a patient with



*ImageCLEFmedCaption_2022_train_079339.jpg*
**Ground truth:** ct chest on presentation
**Prediction:** chest ctscan at disappearance of lung lesion chest computed tomography scan pneumomediastinum with no evidence



*ImageCLEFmedCaption_2022_train_083141.jpg*
**Ground truth:** mri brain image show lt thalamic hyperintensity transverse weight mri section show hyperintensity and mark heterogeneity of signal in a left frontal lobe oligodendroglioma note extension of the lesion through the corpus callosum to the right frontal lobe arrow create a butterfly glioma appearance
**Prediction:** Here we present a series of brain scan images of a patient with

# Bibliography

[1] Abouelmehdi, Karim, Beni-Hssane, Abderrahim, Khaloufi, Hayat, and Saadi, Mostafa. “Big data security and privacy in healthcare: A Review”. In: *Procedia Computer Science* 113 (2017). The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017) / The 7th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2017) / Affiliated Workshops, pp. 73–80. ISSN: 1877-0509. DOI: `https://doi.org/10.1016/j.procs.2017.08.292`. URL: `https://www.sciencedirect.com/science/article/pii/S1877050917317015`.

[2] Alammar, Jay. *The Illustrated Transformer*. URL: `http://jalammar.github.io/illustrated-transformer/`

[3] Anderson, Peter, Fernando, Basura, Johnson, Mark, and Gould, Stephen. “SPICE: Semantic Propositional Image Caption Evaluation”. In: *CoRR* abs/1607.08822 (2016). arXiv: `1607.08822`. URL: `http://arxiv.org/abs/1607.08822`.

[4] Athanasiadis, Ioannis, Moschovis, Georgios, and Tuoma, Alexander. “Weakly-Supervised Semantic Segmentation via Transformer Explainability”. In: *ML Reproducibility Challenge 2021 (Fall Edition)*. 2022. URL: `https://openreview.net/forum?id=rcEDhGX3AY`.

[5] Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: `1409.0473 [cs.CL]`.

[6] Banerjee, Satanjeev and Lavie, Alon. “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”. In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor,

Michigan: Association for Computational Linguistics, June 2005, pp. 65–72. URL: `https://aclanthology.org/W05-0909`.

[7] Charalampakos, Foivos. "Exploring Deep Learning Methods for Medical Image Tagging". MA thesis. Athens, Greece: Athens University of Economics and Business, Sept. 2022.

[8] Charalampakos, Foivos, Zachariadis, Giorgos, Pavlopoulos, John, Karatzas, Vasilis, Trakas, Christoforos, and Androutsopoulos, Ion. "AUEB NLP Group at ImageCLEFmedical Caption 2022". In: *CLEF2022 Working Notes*. CEUR Workshop Proceedings. Bologna, Italy: CEUR-WS.org, Sept. 2022.

[9] Cheng, Jianpeng and Lapata, Mirella. "Neural Summarization by Extracting Sentences and Words". In: *CoRR* abs/1603.07252 (2016). arXiv: `1603.07252`. URL: `http://arxiv.org/abs/1603.07252`.

[10] Cho, Kyunghyun, Merrienboer, Bart van, Gülçehre, Çaglar, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *CoRR* abs/1406.1078 (2014). arXiv: `1406.1078`. URL: `http://arxiv.org/abs/1406.1078`.

[11] Colah, Christopher. *Understanding LSTM Networks*. URL: `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`

[12] Cuenat, Stéphane and Couturier, Raphaël. "Convolutional Neural Network (CNN) vs Visual Transformer (ViT) for Digital Holography". In: *CoRR* abs/2108.09147 (2021). arXiv: `2108.09147`. URL: `https://arxiv.org/abs/2108.09147`.

[13] Darken, Christian J. and Moody, John E. "Note on Learning Rate Schedules for Stochastic Optimization". In: *NIPS*. 1990.

[14] Demner-Fushman, Dina, Kohli, Marc D., Rosenman, Marc B., Shooshan, Sonya E., Rodriguez, Laritza, Antani, Sameer, Thoma, George R., and McDonald, Clement J. English (US). In: *Journal of the American Medical Informatics Association : JAMIA* 23.2 (Mar. 2016). Publisher Copyright: © 2015 Published by Oxford University Press on behalf of the American Medical Informatics Association 2015. This work is written by US Government employees and is in the public domain in the US., pp. 304–310. ISSN: 1067-5027. DOI: `10.1093/jamia/ocv080`.

[15] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: `10.18653/v1/N19-1423`. URL: `https://aclanthology.org/N19-1423`.

[16] Donahue, Jeff, Hendricks, Lisa Anne, Guadarrama, Sergio, Rohrbach, Marcus, Venugopalan, Subhashini, Saenko, Kate, and Darrell, Trevor. "Long-term Recurrent Convolutional Networks for Visual Recognition and Description". In: *CoRR* abs/1411.4389 (2014). arXiv: `1411.4389`. URL: `http://arxiv.org/abs/1411.4389`.

[17] Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, Uszkoreit, Jakob, and Houlsby, Neil. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. URL: `https://openreview.net/forum?id=YicbFdNTTy`.

[18] Eickhoff, Carsten, Schwall, Immanuel, García Seco de Herrera, Alba, and Müller, Henning. "Overview of ImageCLEFcaption 2017 – Image Caption Prediction and Concept Detection for Biomedical Images". In: Jan. 2017.

[19] El Hihi, Salah and Bengio, Yoshua. "Hierarchical Recurrent Neural Networks for Long-Term Dependencies". In: NIPS'95. Denver, Colorado: MIT Press, 1995, pp. 493–499.

[20] Gale, William, Oakden-Rayner, Luke, Carneiro, Gustavo, Bradley, Andrew P., and Palmer, Lyle J. "Producing radiologist-quality reports for interpretable artificial intelligence". In: *CoRR* abs/1806.00340 (2018). arXiv: `1806.00340`. URL: `http://arxiv.org/abs/1806.00340`.

[21] García Seco de Herrera, Alba, Eickhoff, Carsten, Andrearczyk, Vincent, and Müller, Henning. "Overview of the ImageCLEF 2018 caption prediction tasks". In: Sept. 2018.

[22] Glorot, Xavier and Bengio, Yoshua. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Yee Whye Teh and Mike Titterington. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. URL: `https://proceedings.mlr.press/v9/glorot10a.html`.

[23] Graves, Alex. *Generating Sequences With Recurrent Neural Networks*. 2014. arXiv: `1308.0850 [cs.NE]`.

[24] Hajihosseini, Malihe, Lotfollahi, Yasaman, Nobakhtian, Melika, Mahdi Javid, Mohammad, Omidi, Fateme, and Eetemadi, Sauleh. “IUST$_{N}LPLABatImageCLEFmedicalCaptionTasks2022$”. In: *CLEF2022 Working Notes*. CEUR Workshop Proceedings. Bologna, Italy: CEUR-WS.org, Sept. 2022.

[25] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: `10.1109/CVPR.2016.90`.

[26] Hochreiter, Sepp and Schmidhuber, Jürgen. “Long Short-term Memory”. In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: `10.1162/neco.1997.9.8.1735`.

[27] Huang, Gao, Liu, Zhuang, Van Der Maaten, Laurens, and Weinberger, Kilian Q. “Densely Connected Convolutional Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269. DOI: `10.1109/CVPR.2017.243`.

[28] Ionescu, Bogdan, Müller, Henning, Peteri, Renaud, Rückert, Johannes, Abacha, Asma Ben, Herrera, Alba Garcia Seco de, Friedrich, Christoph M., Bloch, Louise, Brüngel, Raphael, Idrissi-Yaghir, Ahmad, Schäfer, Henning, Kozlovski, Serge, Cid, Yashin Dicente, Kovalev, Vassili, Stefan, Liviu-Daniel, Constantin, Mihai Gabriel, Dogariu, Mihai, Popescu, Adrian, Deshayes-Chossart, Jerome, Schindler, Hugo, Chamberlain, Jon, Campello, Antonio, and Clark, Adrian. “Overview of the ImageCLEF 2022: Multimedia Retrieval in Medical, Social Media and Nature Applications”. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Proceedings of the 13th

International Conference of the CLEF Association (CLEF 2022). Bologna, Italy: LNCS Lecture Notes in Computer Science, Springer, Sept. 2022.

[29] Irvin, Jeremy, Rajpurkar, Pranav, Ko, Michael, Yu, Yifan, Ciurea-Ilcus, Silviana, Chute, Chris, Marklund, Henrik, Haghgoo, Behzad, Ball, Robyn L., Shpanskaya, Katie S., Seekins, Jayne, Mong, David A., Halabi, Safwan S., Sandberg, Jesse K., Jones, Ricky, Larson, David B., Langlotz, Curtis P., Patel, Bhavik N., Lungren, Matthew P., and Ng, Andrew Y. "CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison". In: *CoRR* abs/1901.07031 (2019). arXiv: `1901.07031`. URL: `http://arxiv.org/abs/1901.07031`.

[30] Jing, Baoyu, Xie, Pengtao, and Xing, Eric P. "On the Automatic Generation of Medical Imaging Reports". In: *CoRR* abs/1711.08195 (2017). arXiv: `1711.08195`. URL: `http://arxiv.org/abs/1711.08195`.

[31] Johnson, Alistair E. W., Pollard, Tom J., Berkowitz, Seth J., Greenbaum, Nathaniel R., Lungren, Matthew P., Deng, Chih-ying, Mark, Roger G., and Horng, Steven. "MIMIC-CXR: A large publicly available database of labeled chest radiographs". In: *CoRR* abs/1901.07042 (2019). arXiv: `1901.07042`. URL: `http://arxiv.org/abs/1901.07042`.

[32] Johnson, Jeff, Douze, Matthijs, and Jégou, Hervé. "Billion-scale similarity search with GPUs". In: *IEEE Transactions on Big Data* 7.3 (2019), pp. 535–547.

[33] Jonsson, Fredrik. "Evaluation of the Transformer Model for Abstractive Text Summarization". MA thesis. Stockholm, Sweden: KTH Royal Institute of Technology, Aug. 2019.

[34] Karpukhin, Vladimir, Oguz, Barlas, Min, Sewon, Lewis, Patrick, Wu, Ledell, Edunov, Sergey, Chen, Danqi, and Yih, Wen-tau. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: `10.18653/v1/2020.emnlp-main.550`. URL: `https://aclanthology.org/2020.emnlp-main.550`.

[35] Kassner, Nora and Schütze, Hinrich. "Negated LAMA: Birds cannot fly". In: *CoRR* abs/1911.03343 (2019). arXiv: `1911.03343`. URL: `http://arxiv.org/abs/1911.03343`.

[36] Khandelwal, Urvashi, Levy, Omer, Jurafsky, Dan, Zettlemoyer, Luke, and Lewis, Mike. “Generalization through Memorization: Nearest Neighbor Language Models”. In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=HklBjCEKvH`.

[37] Kingma, Diederik and Ba, Jimmy. “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations* (Dec. 2014).

[38] Kisilev, Pavel, Sason, Eli, Barkan, Ella, and Hashoul, Sharbell Y. “Medical image captioning : learning to describe medical image findings using multitask-loss CNN”. In: 2016.

[39] Konar, Jinia, Khandelwal, Prerit, and Tripathi, Rishabh. “Comparison of Various Learning Rate Scheduling Techniques on Convolutional Neural Network”. In: *2020 IEEE International Students’ Conference on Electrical,Electronics and Computer Science (SCEECS)*. 2020, pp. 1–5. DOI: `10.1109/SCEECS48394.2020.94`.

[40] Krizhevsky, Alex. “One weird trick for parallelizing convolutional neural networks”. In: *CoRR* abs/1404.5997 (2014). arXiv: `1404.5997`. URL: `http://arxiv.org/abs/1404.5997`.

[41] Lawson, Christopher E., Martí, Jose Manuel, Radivojevic, Tijana, Jonnalagadda, Sai Vamshi R., Gentz, Reinhard, Hillson, Nathan J., Peisert, Sean, Kim, Joonhoon, Simmons, Blake A., Petzold, Christopher J., Singer, Steven W., Mukhopadhyay, Aindrila, Tanjore, Deepti, Dunn, Joshua G., and Garcia Martin, Hector. “Machine learning for metabolic engineering: A review”. In: *Metabolic Engineering* 63 (2021). Tools and Strategies of Metabolic Engineering, pp. 34–60. ISSN: 1096-7176. DOI: `https://doi.org/10.1016/j.ymben.2020.10.005`. URL: `https://www.sciencedirect.com/science/article/pii/S109671762030166X`.

[42] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: `10.1109/5.726791`.

[43] Lee, Jinhyuk, Yoon, Wonjin, Kim, Sungdong, Kim, Donghyeon, Kim, Sunkyu, So, Chan Ho, and Kang, Jaewoo. “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”. In: *Bioinformatics*

36.4 (Sept. 2019), pp. 1234–1240. ISSN: 1367-4803. DOI: `10.1093/bioinformatics/btz682`.

[44] Lewis, Mike, Liu, Yinhan, Goyal, Naman, Ghazvininejad, Marjan, Mohamed, Abdelrahman, Levy, Omer, Stoyanov, Veselin, and Zettlemoyer, Luke. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 7871–7880. DOI: `10.18653/v1/2020.acl-main.703`. URL: `https://aclanthology.org/2020.acl-main.703`.

[45] Lewis, Patrick, Perez, Ethan, Piktus, Aleksandra, Petroni, Fabio, Karpukhin, Vladimir, Goyal, Naman, Küttler, Heinrich, Lewis, Mike, Yih, Wen-tau, Rocktäschel, Tim, Riedel, Sebastian, and Kiela, Douwe. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. URL: `https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf`.

[46] Li, Christy Y., Liang, Xiaodan, Hu, Zhiting, and Xing, Eric P. “Hybrid Retrieval-Generation Reinforced Agent for Medical Image Report Generation”. In: *CoRR* abs/1805.08298 (2018). arXiv: `1805.08298`. URL: `http://arxiv.org/abs/1805.08298`.

[47] Li, Christy Y., Liang, Xiaodan, Hu, Zhiting, and Xing, Eric P. “Knowledge-driven Encode, Retrieve, Paraphrase for Medical Image Report Generation”. In: *CoRR* abs/1903.10122 (2019). arXiv: `1903.10122`. URL: `http://arxiv.org/abs/1903.10122`.

[48] Li, Pengfei, Zhong, Peixiang, Mao, Kezhi, Wang, Dongzhe, Yang, Xuefeng, Liu, Yunfeng, Yin, Jianxiong, and See, Simon. “ACT: an Attentive Convolutional Transformer for Efficient Text Classification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.15 (May 2021), pp. 13261–13269. URL: `https://ojs.aaai.org/index.php/AAAI/article/view/17566`.

[49] Lin, Chin-Yew. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches*

*Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: `https://aclanthology.org/W04-1013`.

[50] Liu, Guanxiong, Hsu, Tzu-Ming Harry, McDermott, Matthew B. A., Boag, Willie, Weng, Wei-Hung, Szolovits, Peter, and Ghassemi, Marzyeh. "Clinically Accurate Chest X-Ray Report Generation". In: *CoRR* abs/1904.02633 (2019). arXiv: `1904.02633`. URL: `http://arxiv.org/abs/1904.02633`.

[51] Liu, Zhuang, Mao, Hanzi, Wu, Chao-Yuan, Feichtenhofer, Christoph, Darrell, Trevor, and Xie, Saining. "A ConvNet for the 2020s". In: *CoRR* abs/2201.03545 (2022). arXiv: `2201.03545`. URL: `https://arxiv.org/abs/2201.03545`.

[52] Loshchilov, Ilya and Hutter, Frank. *Fixing Weight Decay Regularization in Adam*. 2018. URL: `https://openreview.net/forum?id=rk6qdGgCZ`.

[53] MacQueen, J. B. "Some Methods for Classification and Analysis of MultiVariate Observations". In: *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. Ed. by L. M. Le Cam and J. Neyman. Vol. 1. University of California Press, 1967, pp. 281–297.

[54] Molnar, Christoph. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. URL: `https://christophm.github.io/interpretable-ml-book`.

[55] Moschovis, Georgios and Fransén, Erik. "NeuralDynamicsLab at ImageCLEF Medical 2022". In: *CLEF2022 Working Notes*. CEUR Workshop Proceedings. Bologna, Italy: CEUR-WS.org, Sept. 2022.

[56] Mussmann, Stephen and Ermon, Stefano. "Learning and Inference via Maximum Inner Product Search". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2587–2596. URL: `https://proceedings.mlr.press/v48/mussmann16.html`.

[57] Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: `10.3115/1073083.1073135`. URL: `https://aclanthology.org/P02-1040`.

[58] Pavlopoulos, John, Kougia, Vasiliki, and Androutsopoulos, Ion. “A Survey on Biomedical Image Captioning”. In: *Proceedings of the Second Workshop on Shortcomings in Vision and Language*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 26–36. DOI: `10.18653/v1/W19-1803`. URL: `https://aclanthology.org/W19-1803`.

[59] Pavlopoulos, John, Kougia, Vasiliki, Androutsopoulos, Ion, and Papamichail, Dimitris. “Diagnostic captioning: a survey”. In: *Knowledge and Information Systems* (June 2022), pp. 1–32. DOI: `10.1007/s10115-022-01684-7`.

[60] Pelka, Obioma, Koitka, Sven, Rückert, Johannes, Nensa, Felix, and Friedrich, Christoph M. “Radiology Objects in COntext (ROCO): A Multimodal Image Dataset”. In: *Intravascular Imaging and Computer Assisted Stenting - and - Large-Scale Annotation of Biomedical Data and Expert Label Synthesis - 7th Joint International Workshop, CVII-STENT 2018 and Third International Workshop, LABELS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings*. Ed. by Danail Stoyanov, Zeike Taylor, Simone Balocco, Raphael Sznitman, Anne L. Martel, Lena Maier-Hein, Luc Duong, Guillaume Zahnd, Stefanie Demirci, Shadi Albarqouni, Su-Lin Lee, Stefano Moriconi, Veronika Cheplygina, Diana Mateus, Emanuele Trucco, Eric Granger, and Pierre Jannin. Vol. 11043. Lecture Notes in Computer Science. Springer, 2018, pp. 180–189. DOI: `10.1007/978-3-030-01364-6\_20`. URL: `https://doi.org/10.1007/978-3-030-01364-6%5C_20`.

[61] Peng, Yifan, Yan, Shankai, and Lu, Zhiyong. “Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets”. In: *CoRR* abs/1906.05474 (2019). arXiv: `1906.05474`. URL: `http://arxiv.org/abs/1906.05474`.

[62] Pilault, Jonathan, Li, Raymond, Subramanian, Sandeep, and Pal, Chris. “On Extractive and Abstractive Neural Document Summarization with Transformer Language Models”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 9308–9319. DOI: `10.18653/v1/2020.emnlp-main.748`. URL: `https://aclanthology.org/2020.emnlp-main.748`.

[63] Radenović, Filip, Tolias, Giorgos, and Chum, Ondřej. “Fine-Tuning CNN Image Retrieval with No Human Annotation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.7 (2019), pp. 1655–1668. DOI: `10.1109/TPAMI.2018.2846566`.

[64] Rajpurkar, Pranav, Irvin, Jeremy, Zhu, Kaylie, Yang, Brandon, Mehta, Hershel, Duan, Tony, Ding, Daisy Yi, Bagul, Aarti, Langlotz, Curtis P., Shpanskaya, Katie S., Lungren, Matthew P., and Ng, Andrew Y. “CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning”. In: *CoRR* abs/1711.05225 (2017). arXiv: `1711.05225`. URL: `http://arxiv.org/abs/1711.05225`.

[65] Ren, Pengzhen, Xiao, Yun, Chang, Xiaojun, Huang, Po-Yao, Li, Zhihui, Gupta, Brij B., Chen, Xiaojiang, and Wang, Xin. *A Survey of Deep Active Learning*. 2021. arXiv: `2009.00236 [cs.LG]`.

[66] Rennie, Steven J., Marcheret, Etienne, Mroueh, Youssef, Ross, Jerret, and Goel, Vaibhava. “Self-critical Sequence Training for Image Captioning”. In: *CoRR* abs/1612.00563 (2016). arXiv: `1612.00563`. URL: `http://arxiv.org/abs/1612.00563`.

[67] Robbins, H. and Monro, S. “A stochastic approximation method”. In: *Annals of Mathematical Statistics* 22 (1951), pp. 400–407.

[68] Rückert, Johannes, Ben Abacha, Asma, García Seco de Herrera, Alba, Bloch, Louise, Brüngel, Raphael, Idrissi-Yaghir, Ahmad, Schäfer, Henning, Müller, Henning, and Friedrich, Christoph M. “Overview of ImageCLEFmedical 2022 – Caption Prediction and Concept Detection”. In: *CLEF2022 Working Notes*. CEUR Workshop Proceedings. Bologna, Italy: CEUR-WS.org, Sept. 2022.

[69] Rush, Alexander. *The Annotated Transformer*.

[70] Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander, and Fei-Fei, Li. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115 (Sept. 2014). DOI: `10.1007/s11263-015-0816-y`.

[71] Schlegl, Thomas, Waldstein, Sebastian, Vogl, Wolf-Dieter, Schmidt-Erfurth, Ursula, and Langs, Georg. “Predicting Semantic Descriptions from Medical Images with Convolutional Neural Networks”. In: vol. 24. July 2015, pp. 437–48. ISBN: 978-3-319-19991-7. DOI: `10.1007/978-3-319-19992-4_34`.

[72] Shin, Hoo-Chang, Roberts, Kirk, Lu, Le, Demner-Fushman, Dina, Yao, Jianhua, and Summers, Ronald M. “Learning to Read Chest X-Rays: Recurrent Neural Cascade Model for Automated Image Annotation”. In: *CoRR* abs/1603.08486 (2016). arXiv: `1603.08486`. URL: `http://arxiv.org/abs/1603.08486`.

[73] Simonyan, Karen and Zisserman, Andrew. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *arXiv 1409.1556* (Sept. 2014).

[74] Smith, Leslie N. “No More Pesky Learning Rate Guessing Games”. In: *CoRR* abs/1506.01186 (2015). arXiv: `1506.01186`. URL: `http://arxiv.org/abs/1506.01186`.

[75] Sullivan, Gail and Feinn, Richard. “Using Effect Size—or Why the P Value Is Not Enough”. In: *Journal of graduate medical education* 4 (Sept. 2012), pp. 279–82. DOI: `10.4300/JGME-D-12-00156.1`.

[76] Surowiecki, James. *The Wisdom of Crowds*. Anchor, 2005. ISBN: 0385721706.

[77] Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott E., Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. “Going Deeper with Convolutions”. In: *CoRR* abs/1409.4842 (2014). arXiv: `1409.4842`. URL: `http://arxiv.org/abs/1409.4842`.

[78] Tan, Mingxing and Le, Quoc V. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *CoRR* abs/1905.11946 (2019). arXiv: `1905.11946`. URL: `http://arxiv.org/abs/1905.11946`.

[79] Tsochantaridis, Ioannis, Hofmann, Thomas, Joachims, Thorsten, and Altun, Yasemin. “Support Vector Machine Learning for Interdependent and Structured Output Spaces”. In: *Machine Learning* (July 2004). DOI: `10.1145/1015330.1015341`.

[80] Varges, Sebastian, Bieler, Heike, Stede, Manfred, Faulstich, Lukas C., Irsig, Kristin, and Atalla, Malik. “SemScribe: Natural Language Generation for Medical Reports”. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*. Istanbul, Turkey: European

Language Resources Association (ELRA), May 2012, pp. 2674–2681. URL: `http://www.lrec-conf.org/proceedings/lrec2012/pdf/165_Paper.pdf`.

[81] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, and Polosukhin, Illia. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[82] Vedantam, Ramakrishna, Zitnick, C. Lawrence, and Parikh, Devi. “CIDEr: Consensus-based Image Description Evaluation”. In: *CoRR* abs/1411.5726 (2014). arXiv: `1411.5726`. URL: `http://arxiv.org/abs/1411.5726`.

[83] Vincent, Pascal, Larochelle, Hugo, Bengio, Yoshua, and Manzagol, Pierre-Antoine. “Extracting and Composing Robust Features with Denoising Autoencoders”. In: *Proceedings of the 25th International Conference on Machine Learning*. ICML ’08. Helsinki, Finland: Association for Computing Machinery, 2008, pp. 1096–1103. ISBN: 9781605582054. DOI: `10.1145/1390156.1390294`. URL: `https://doi.org/10.1145/1390156.1390294`.

[84] Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. “Show and Tell: A Neural Image Caption Generator”. In: *CoRR* abs/1411.4555 (2014). arXiv: `1411.4555`. URL: `http://arxiv.org/abs/1411.4555`.

[85] Wang, Hongyu and Xia, Yong. “ChestNet: A Deep Neural Network for Classification of Thoracic Diseases on Chest Radiography”. In: *CoRR* abs/1807.03058 (2018). arXiv: `1807.03058`. URL: `http://arxiv.org/abs/1807.03058`.

[86] Wang, Xiaosong, Peng, Yifan, Lu, Le, Lu, Zhiyong, Bagheri, Mohammadhadi, and Summers, Ronald M. “ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases”. In: *CoRR* abs/1705.02315 (2017). arXiv: `1705.02315`. URL: `http://arxiv.org/abs/1705.02315`.

[87] Wang, Xiaosong, Peng, Yifan, Lu, Le, Lu, Zhiyong, and Summers, Ronald M. “TieNet: Text-Image Embedding Network for Common Thorax Disease

Classification and Reporting in Chest X-rays”. In: *CoRR* abs/1801.04334 (2018). arXiv: `1801.04334`. URL: `http://arxiv.org/abs/1801.04334`.

[88] Williams, Ronald J. “Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning”. In: *Mach. Learn.* 8.3–4 (May 1992), pp. 229–256. ISSN: 0885-6125. DOI: `10.1007/BF00992696`. URL: `https://doi.org/10.1007/BF00992696`.

[89] Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron C., Salakhutdinov, Ruslan, Zemel, Richard S., and Bengio, Yoshua. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”. In: *CoRR* abs/1502.03044 (2015). arXiv: `1502.03044`. URL: `http://arxiv.org/abs/1502.03044`.

[90] You, Quanzeng, Jin, Hailin, Wang, Zhaowen, Fang, Chen, and Luo, Jiebo. *Image Captioning with Semantic Attention*. 2016. arXiv: `1603.03925 [cs.CV]`.

[91] Zhang, Jingqing, Zhao, Yao, Saleh, Mohammad, and Liu, Peter J. “PEGASUS: Pre-Training with Extracted Gap-Sentences for Abstractive Summarization”. In: *Proceedings of the 37th International Conference on Machine Learning*. ICML’20. JMLR.org, 2020.

[92] Zhang, Wei, Tanida, Jun, Itoh, Kazuyoshi, and Ichioka, Yoshiki. “Shift-invariant pattern recognition neural network and its optical architecture”. In: *Proceedings of annual conference of the Japan Society of Applied Physics*. Montreal, CA. 1988, pp. 2147–2151.

[93] Zhang, Yuhao, Merck, Derek, Tsai, Emily Bao, Manning, Christopher D., and Langlotz, Curtis P. “Optimizing the Factual Correctness of a Summary: A Study of Summarizing Radiology Reports”. In: *CoRR* abs/1911.02541 (2019). arXiv: `1911.02541`. URL: `http://arxiv.org/abs/1911.02541`.

[94] Zhang, Zizhao, Chen, Pingjun, Sapkota, Manish, and Yang, Lin. *TandemNet: Distilling Knowledge from Medical Images Using Diagnostic Reports as Optional Semantic References*. 2017. arXiv: `1708.03070 [cs.CV]`.

[95] Zhang, Zizhao, Xie, Yuanpu, Xing, Fuyong, McGough, Mason, and Yang, Lin. *MDNet: A Semantically and Visually Interpretable Medical Image Diagnosis Network*. 2017. arXiv: `1707.02485 [cs.CV]`.

# Chapter 6

# Experimental results and details

## 6.1 ImageCLEFmedical 2022 contributions

In this part, we present the results of our algorithms proposed in ImageCLEFmedical 2022 evaluation campaign, in order of performance, for both its concept detection and caption generation subtasks that rely on pre-trained convolutional encoders, which are extremely popular in computer vision. In this part we include the $F_1$ scores in all of training, validation, test sets in which each proposed baseline has been evaluated. My proposed systems, on behalf of *NeuralDynamicsLab*, ranked $4^{\text{th}}$ in the former and $5^{\text{th}}$ in the latter subtask [55].

As mentioned in section 3.1.1 after merging the initially provided train and validation data, we shuffle them after manually setting the seeds to eliminate randomness in consecutive runs while tuning our hyperparameters and then keep $80\%$ as our training set, $10\%$ as our validation set to perform hyperparameter tuning and the remaining $10\%$ as our development set to perform model selection. Since the dataset is large we perform neither cross-validation nor data-augmentation but set all the random seeds equal to $0$, the CUDNNs backends as deterministic and disable the CUDNNs backends benchmark in all runs.

### 6.1.1 Concept Detection Performance summary

Table 6.1.1 below summarizes several characteristics of the proposed baselines for the concept detection task, in order of performance with respect to $F_1$ scores, together with their descriptions' locations in chapter 4. We observe that DenseNet161 encoders with finetuned classification heads are our top performing configurations and outperform other CNN architectures, which is in accordance with their extensive use for X-RAYs processing [64]. In contrast, fully finetuning the backbone networks and then using retrieval based heuristics that capture their representations' embeddings similarities, such as the 1-NN baseline [50], achieve lower scores. Additional details and $F_1$ scores for all our configurations are provided in chapter 7 (Appendix).

Table 6.1.1: Summary of our configurations' training targets and $F_1$ scores

| Section | Table | Training target | Dev. $F_1$ | Val. $F_1$ | Test $F_1$ |
|---|---|---|---|---|---|
| Section 4.1.1(a) | - | DenseNet161 Head | 0.44460 | 0.44614 | 0.43601 |
| Section 4.1.1(a) | - | DenseNet161 Head | 0.44460 | 0.44614 | 0.43567 |
| Section 4.1.1(b) | - | DenseNet161 Head | 0.44429 | 0.44516 | 0.43558 |
| Section 4.1.1(c) | - | DenseNet161 Head | 0.44430 | 0.44524 | 0.43539 |
| Section 4.1.1(d) | Table 4.1.1 | Ensemble of DenseNets | 0.44544 | 0.44553 | 0.43496 |
| Section 4.1.1(e) | Table 4.1.2 | Ensemble of Networks | 0.44170 | 0.44167 | 0.43404 |
| Section 4.1.1(e) | Table 4.1.3 | Ensemble of Networks | 0.44305 | 0.44379 | 0.43130 |
| Section 4.1.1(e) | Table 4.1.4 | Ensemble of Networks | 0.44543 | 0.44623 | 0.42957 |
| Section 4.1.1(f) | - | DenseNet161 (finetuned) | 0.32418 | 0.32654 | 0.31687 |
| Section 4.1.1(g) | - | VGG-16 NN search | 0.25202 | 0.25276 | 0.25061 |

### 6.1.2 Caption Generation Performance summary

Tables 6.1.2, 6.1.2, 6.1.4 below present some characteristics of the employed baselines for the caption generation task, in order of performance with respect to BLEU scores, together with their descriptions' locations in chapter 4. We observe that although RAG models perform better than solely the 1-NN baseline, when the latter is combined with abstractive summarization techniques for the diagnostic texts, it performs better with a carefully selected image encoder.

Table 6.1.2: Summary of our configurations' parameters and IDs

| ID | Section | Table | Encoder | Generator | Captions $k$ | Tokens $n$ |
|---|---|---|---|---|---|---|
| DC01 | Section 4.1.2(a) | Table 4.1.5 | AlexNet | Pegasus | $k = 9$ | $n = 15$ |
| DC02 | Section 4.1.2(a) | Table 4.1.5 | AlexNet | Pegasus | $k = 4$ | $n = 15$ |
| DC03 | Section 4.1.2(a) | Table 4.1.5 | AlexNet | Pegasus | $k = 3$ | $n = 15$ |
| DC04 | Section 4.1.2(a) | Table 4.1.5 | AlexNet | Pegasus | $k = 2$ | $n = 15$ |
| DC05 | Section 4.1.2(a) | Table 4.1.5 | AlexNet | Pegasus | $k = 4$ | $n = 5$ |
| DC06 | Section 4.1.2(a) | Table 4.1.5 | AlexNet | Pegasus | $k = 3$ | $n = 5$ |
| DC07 | Section 4.1.2(b) | Table 4.1.6 | AlexNet | RAG | $k = 1$ | - |
| DC08 | Section 4.1.2(b) | Table 4.1.6 | VGG-16 | RAG | $k = 1$ | - |
| DC09 | Section 4.1.2(c) | Table 4.1.7 | AlexNet | 1-NN | $k = 1$ | - |
| DC10 | Section 4.1.2(c) | Table 4.1.7 | VGG-16 | 1-NN | $k = 1$ | - |

Table 6.1.3: Summary of our configurations' test scores in all ImageCLEF metrics

| ID | BLEU | ROUGE-L | METEOR | CIDER | SPICE | BERTscore |
|---|---|---|---|---|---|---|
| DC01 | 0.29166 | 0.11566 | 0.06240 | 0.13169 | 0.02182 | 0.57282 |
| DC02 | 0.28343 | 0.11128 | 0.05946 | 0.13409 | 0.02072 | 0.57338 |
| DC03 | 0.27855 | 0.11101 | 0.05828 | 0.13963 | 0.02193 | 0.57427 |
| DC04 | 0.27007 | 0.11031 | 0.05675 | 0.14999 | 0.02270 | 0.57579 |
| DC05 | 0.25521 | 0.11373 | 0.05485 | 0.17865 | 0.02556 | 0.57648 |
| DC06 | 0.25334 | 0.11246 | 0.05407 | 0.17907 | 0.025401 | 0.57669 |
| DC07 | 0.25127 | 0.10528 | 0.05200 | 0.15899 | 0.02295 | 0.57337 |
| DC08 | 0.23958 | 0.08808 | 0.04373 | 0.09766 | 0.01607 | 0.56664 |
| DC09 | 0.24064 | 0.11101 | 0.05190 | 0.018900 | 0.02808 | 0.57889 |
| DC10 | 0.22757 | 0.09221 | 0.04353 | 0.11408 | 0.01854 | 0.57059 |

Table 6.1.4: Summary of our configurations' validation scores in various metrics

| ID | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| DC01 | 0.148 | 0.071 | 0.039 | 0.250 | 0.140 | 0.074 | 0.090 |
| DC02 | 0.155 | 0.074 | 0.041 | 0.270 | 0.139 | 0.069 | 0.123 |
| DC03 | 0.157 | 0.077 | 0.045 | 0.310 | 0.138 | 0.069 | 0.129 |
| DC04 | 0.156 | 0.073 | 0.041 | 0.270 | 0.136 | 0.064 | 0.137 |
| DC05 | 0.158 | 0.076 | 0.043 | 0.280 | 0.136 | 0.065 | 0.149 |
| DC06 | 0.156 | 0.074 | 0.041 | 0.280 | 0.136 | 0.064 | 0.140 |
| DC07 | 0.170 | 0.087 | 0.052 | 0.350 | 0.152 | 0.072 | 0.193 |
| DC08 | 0.193 | 0.063 | 0.032 | 0.210 | 0.119 | 0.052 | 0.095 |
| DC09 | 0.156 | 0.076 | 0.043 | 0.300 | 0.135 | 0.061 | 0.166 |
| DC10 | 0.144 | 0.065 | 0.033 | 0.200 | 0.120 | 0.054 | 0.104 |

### Ablation study on abstractive summarization baseline

As mentioned above our best performing captioning models have extended the simple 1-NN baseline [50] for caption generation. Precisely, 1-NN generation constitutes the first part of our models' generated caption, however apart from the neighbour with the closest representation, we retrieve the top-$(k+1)$ nearest neighbours, concatenate their outputs, excluding that of the most similar image and feed them as input to an abstractive summarizer; Pegasus [91]. When 1-NN is combined with Pegasus outputs for the diagnostic texts of $k$ additional visually similar images from the training set, where $k \in \mathbb{Z}^+$, it achieves better results on the test set.

To highlight the importance of the concatenation and thus of the 1-NN baseline, we performed an ablation study for several of the respective baselines, where we only use Pegasus summaries on the top-$(k+1)$ nearest neighbours, including the most similar one. The BLEU validation scores are lower as expecte therefore the 1-NN plays a crucial role in the evaluation outcomes.

Table 6.1.5: Summary of our modifications' validation scores in various metrics

| Captions | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| $k = 9$ | 0.069 | 0.031 | 0.015 | 0.100 | 0.087 | 0.52 | 0.020 |
| $k = 4$ | 0.090 | 0.040 | 0.018 | 0.100 | 0.092 | 0.047 | 0.026 |
| $k = 3$ | 0.100 | 0.041 | 0.018 | 0.100 | 0.089 | 0.042 | 0.026 |
| $k = 2$ | 0.110 | 0.046 | 0.022 | 0.120 | 0.088 | 0.039 | 0.025 |
| $k = 1$ | 0.097 | 0.038 | 0.017 | 0.100 | 0.080 | 0.030 | 0.023 |

### Model selection: Image encoders performance on the 1-NN baseline

As it may be realized by the results above, the best performing backbone network for the caption generation subtask is AlexNet, followed by VGG-16. This derivation can be further confirmed by the performance obtained in the validation set, that is taken advantage of for model selection. These are the backbone network architectures used in our proposed baselines using Pegasus, RAG-token, 1-NN, as we have described them in sections 4.1.2(a), 4.1.2(b), 4.1.2(c) or tables 4.1.5, 4.1.6, 4.1.7 respectively. In addition, we observe that although RAG generator performs better in the validation set, Pegasus summarizer scores better on the test set.

Table 6.1.6: Summary of our 1-NN backbones validation scores in various metrics

| Encoder | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| AlexNet | **0.156** | **0.076** | **0.043** | **0.300** | **0.135** | **0.061** | **0.166** |
| VGG-16 | 0.144 | 0.065 | 0.033 | 0.200 | 0.120 | 0.054 | 0.104 |
| VGG-13 | 0.111 | 0.043 | 0.015 | 0.060 | 0.093 | 0.037 | 0.027 |
| EffNetB3 | 0.072 | 0.018 | 0.005 | 0.020 | 0.071 | 0.034 | 0.015 |
| EffNetB5 | 0.055 | 0.010 | 0.003 | 0.010 | 0.056 | 0.018 | 0.010 |
| ResNet50 | 0.085 | 0.029 | 0.001 | 0.060 | 0.079 | 0.026 | 0.027 |
| ResNet101 | 0.120 | 0.035 | 0.016 | 0.100 | 0.103 | 0.037 | 0.042 |
| DenseNet121 | 0.066 | 0.024 | 0.007 | 0.020 | 0.010 | 0.024 | 0.012 |
| DenseNet161 | 0.088 | 0.032 | 0.010 | 0.030 | 0.078 | 0.031 | 0.016 |

Table 6.1.7: Summary of our 1-NN backbones development scores in various metrics

| Encoder | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| AlexNet | **0.157** | **0.080** | **0.048** | **0.350** | **0.134** | **0.062** | **0.168** |
| VGG-16 | 0.139 | 0.063 | 0.032 | 0.210 | 0.119 | 0.052 | 0.095 |
| VGG-13 | 0.111 | 0.043 | 0.015 | 0.060 | 0.092 | 0.037 | 0.027 |
| EffNetB3 | 0.071 | 0.017 | 0.005 | 0.020 | 0.071 | 0.034 | 0.014 |
| EffNetB5 | 0.055 | 0.010 | 0.003 | 0.000 | 0.054 | 0.018 | 0.011 |
| ResNet50 | 0.084 | 0.029 | 0.001 | 0.050 | 0.078 | 0.026 | 0.021 |
| ResNet101 | 0.119 | 0.033 | 0.014 | 0.080 | 0.101 | 0.037 | 0.032 |
| DenseNet121 | 0.065 | 0.024 | 0.007 | 0.020 | 0.026 | 0.024 | 0.011 |
| DenseNet161 | 0.087 | 0.032 | 0.009 | 0.030 | 0.078 | 0.031 | 0.016 |

#### Other baselines: WinnerTakesAll and Dense Passage Retriever

Apart from the 1-NN baseline, we also compute the validation scores of the simplistic WinnerTakesAll baseline and the Dense Passage Retriever based solely on the images' captions, in order to obtain a lower and an upper bound of the performance scores for the remaining Diagnostic Captioning approaches. The WinnerTakesAll baseline, as it is described in section 4.2.1, uses the words frequency in the training captions and takes them in descending order, s.t. $f_i \geq f_{i+1}$, in order to generate the same caption for all instances of the test set. The Dense Passage Retriever, as it is described in section 3.3.1, performs text retrieval based only on the captions; $\boldsymbol{y}$ and totally omitting the visual features of the X-RAYs; $\mathbf{e}(\boldsymbol{x})$.

Table 6.1.8: Summary of our extra baselines validation scores in various metrics

| Method | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| WTA | 0.073 | 0.06 | 0.000 | 0.000 | 0.064 | 0.089 | 0.000 |
| DPR | 0.242 | 0.141 | 0.093 | 0.068 | 0.253 | 0.123 | 0.607 |

## 6.2 Additional experiments on IU X-RAY

In this section, we present results on the additional expepiments that we performed on IU X-RAY dataset [14], which we also mention in section 4.2, a known and publicly available biomedical Dataset that contains medical images and diagnostic reports, to investigate the consistency of our proposed architectures' performance in different settings including noticeable class imbalance. As also explained in section 3.1.2 we took advantage of both IMPRESSION and FINDINGS fields where they exist; to generate the images' captions and the professional shall consider the remaining fields according to the pipeline in figure 1.5.1.

### 6.2.1 Diagnostic Captioning experiments

Following the exact same recipe as in ImageCLEF Medical, we perform an extensive parameter search for caption detection with the same image encoders minimizing the negative $F_1$ score again, whereas for caption generation we focus on the 1-NN baseline and experiment with different backbone networks, the generalized $k$-NN baseline, as well as ensembles of different encoders. For the concept detection task we consider the human authored tags.

### Image encoders performance on the 1-NN baseline

In the case of IU X-RAY, the best performing backbone network with respect again to BLEU 1-4 that was used to compare baselines for the caption generation subtask is EfficientNetB5, followed by AlexNet that was the top-scoring image encoder for the same task and using the same baseline when measuring performance on the data of ImageCLEF Medical 2022.

Table 6.2.1: Summary of our 1-NN backbones validation scores in various metrics

| Encoder | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| AlexNet | 0.307 | 0.173 | 0.105 | 0.068 | 0.228 | 0.131 | 0.143 |
| VGG-16 | 0.307 | 0.172 | 0.105 | 0.067 | 0.230 | 0.130 | 0.158 |
| VGG-13 | 0.303 | 0.170 | 0.102 | 0.064 | 0.233 | 0.129 | 0.102 |
| EffNetB5 | 0.304 | 0.182 | 0.117 | 0.078 | 0.246 | 0.151 | 0.103 |
| EffNetB7 | 0.265 | 0.130 | 0.067 | 0.033 | 0.187 | 0.130 | 0.062 |
| DenseNet121 | 0.256 | 0.138 | 0.083 | 0.053 | 0.213 | 0.110 | 0.169 |
| Ensemble | 0.306 | 0.173 | 0.106 | 0.069 | 0.227 | 0.130 | 0.147 |

### Tuning the value of $k$ for a fixed image encoder

In addition, we further experiment with different values for the number of neighbours $k \in \mathbb{N}^+$ in the generalized $k$-NN baseline, while we use a fixed ResNet50 backbone network to embed the images. We observe that METEOR and SPICE are proportional to $k$, whereas BLUE, ROUGE_L and CIDEr are inversely proportional to $k$. Moreover, some metrics might be dependent on the predicted passage's length, therefore for the respective experiments in the WinnerTakesAll baseline we trim the frequent words to the average gold caption length that is around 33-37 tokens (depending on the split when performing $5 \times 2$-fold cross validation) for IU X-RAY. An indicative example of using 1-NN and WinnerTakesAll baselines is presented hereunder together with their validation performance.

Table 6.2.2: Summary of our 1-NN backbones validation scores in various metrics

| Neighbours $k$ | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| $k = 1$ | 0.288 | 0.155 | 0.092 | 0.056 | 0.214 | 0.118 | 0.102 |
| $k = 2$ | 0.222 | 0.127 | 0.078 | 0.050 | 0.212 | 0.160 | 0.030 |
| $k = 3$ | 0.179 | 0.108 | 0.069 | 0.045 | 0.196 | 0.175 | 0.005 |
| $k = 4$ | 0.151 | 0.094 | 0.062 | 0.041 | 0.181 | 0.178 | 0.000 |
| $k = 5$ | 0.130 | 0.083 | 0.056 | 0.038 | 0.167 | 0.175 | 0.000 |
| $k = 10$ | 0.077 | 0.053 | 0.038 | 0.027 | 0.120 | 0.145 | 0.000 |

### Indicative example of $5 \times 2$-fold cross validation

For caption generation, we have reproduced the 1-NN Network and WinnerTakesAll (WTA) baselines for the IU X-RAY dataset. Recall that, if we denote by $\mathbf{e}(.)$ the output of the employed image encoder among those mentioned in section 1.5, 1-NN predicts $(\hat{x}, \hat{y}) = (\hat{x}, y^*)$ that satisfies $(x^*, y^*) = \arg\min_{\hat{x}} \cos\left(\mathbf{e}(\hat{x}), \mathbf{e}(x^*)\right)$. Meanwhile, the WTA baseline outputs the words in the training captions in decreasing frequency order and always generates the same caption. The process is based on a $90\%$ train-$10\%$ validation split and, since we do not perform model selection, there is no need for a separate hold-out development set.

In Split 1 there are 6674 captions in total among which 2745 are unique, while in WTA output, we have 251613 tokens. In Split 2 there are 6658 captions in total (2738 unique), while in WTA output, we have 250910 tokens. In Split 3 there are 6694 captions in total (2753 unique), while in WTA output, we have 251135 tokens. In Split 4 there are 6700 captions in total (2751 unique), while in WTA output, we have 253229 tokens. In Split 5 there are 6683 captions in total (2738 unique), while in WTA output, we have 252631 tokens.

Table 6.2.3: Average validation scores for the WinnerTakesAll and the 1-NN baseline

| Network | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| WTA | 0.4336 | 0.0704 | 0.0028 | 0.0000 | 0.1864 | 0.1738 | 0.1580 |
| 1-NN | 0.2852 | 0.1550 | 0.0912 | 0.0548 | 0.2108 | 0.1178 | 0.1000 |

Table 6.2.4: Validation scores per split for the WinnerTakesAll baseline

| Fold ID | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| Fold 1 | 0.4420 | 0.0780 | 0.0000 | 0.0000 | 0.1870 | 0.1760 | 0.1660 |
| Fold 2 | 0.4350 | 0.0700 | 0.0000 | 0.0000 | 0.1870 | 0.1730 | 0.1730 |
| Fold 3 | 0.4260 | 0.0660 | 0.0070 | 0.0000 | 0.1850 | 0.1690 | 0.1530 |
| Fold 4 | 0.4300 | 0.0740 | 0.0000 | 0.0000 | 0.1830 | 0.1740 | 0.1560 |
| Fold 5 | 0.4350 | 0.0640 | 0.0070 | 0.0000 | 0.1900 | 0.1770 | 0.1440 |

Table 6.2.5: Validation scores per split for the 1-NN baseline

| Fold ID | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE-L | METEOR | CIDER |
|---|---|---|---|---|---|---|---|
| Fold 1 | 0.2810 | 0.1530 | 0.0890 | 0.0530 | 0.2050 | 0.1160 | 0.1080 |
| Fold 2 | 0.2880 | 0.1550 | 0.0920 | 0.0560 | 0.2140 | 0.1180 | 0.1020 |
| Fold 3 | 0.2810 | 0.1520 | 0.0890 | 0.0520 | 0.2050 | 0.1140 | 0.0720 |
| Fold 4 | 0.2900 | 0.1590 | 0.0940 | 0.0580 | 0.2130 | 0.1190 | 0.1190 |
| Fold 5 | 0.2860 | 0.1560 | 0.0920 | 0.0550 | 0.2170 | 0.1220 | 0.0970 |

# Chapter 7

# Additional results and statistics

Last but not least, in this part we present details on performing hyper-parameter tuning for the Concept Detection encoders, as well as the complete outcomes of Statistical Significance Tests for all our proposed systems. The $F_1$ scores of our tagging baselines and the statistical significance tests' results are presented hereunder, marked in red in cases where we reject the null hypothesis $H_0$ and green when we fail to do so, while regarding the effect size, we use orange when it is small, blue when it is medium and purple when it is large in order to navigate the reader.

Table 7.1: Summary of all ImageCLEF Concept Prediction experiments configurations' training targets and F1 scores

| TRAINING REGIME | DEEP LEARNING MODEL<br>(Default configuration setting: Adam optimizer, Batch size: 120, neg. F1 loss, grad. clipping, 1-layered head) | VALIDATION F1 LOSS | DEVELOPMENT F1 LOSS |
|---|---|---|---|
| Back-propagation | AlexNet with batch size 60, cyclic lr from $10^{-5}$ to $10^{-1}$ with ns=4, AdamW optimizer | 0,40479 | 0,40368 |
| Back-propagation | AlexNet with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,43604 | 0,43535 |
| Back-propagation | AlexNet with learning rate $10^{-5}$ and batch size 60, AdamW optimizer | 0,42783 | 0,42683 |
| Back-propagation | AlexNet with learning rate $10^{-5}$ and batch size 60, BCE loss, AdamW optimizer | 0,36920 | 0,37162 |
| Back-propagation | AlexNet with learning rate $10^{-5}$ with linear decay and batch size 60, AdamW optimizer | 0,42314 | 0,42156 |
| Back-propagation | AlexNet with learning rate $5 \times 10^{-4}$ and batch size 60, AdamW optimizer | 0,43429 | 0,43374 |
| Back-propagation | AlexNet with learning rate $5 \times 10^{-5}$ and batch size 60, AdamW optimizer | 0,43539 | 0,43432 |
| Back-propagation | AlexNet with learning rate $5 \times 10^{-5}$ with linear decay and batch size 60, AdamW optimizer | 0,43272 | 0,43163 |
| Back-propagation | AlexNet with learning rate $5 \times 10^{-5}$ and batch size 60, 2-layered head with 4187 hidden nodes, AdamW optimizer | 0,43022 | 0,42897 |
| Back-propagation | AlexNet with learning rate $5 \times 10^{-5}$ and batch size 60, noise with probability $25 \times 10^{-2}$, AdamW optimizer | 0,43523 | 0,43351 |
| Back-propagation | AlexNet with learning rate $5 \times 10^{-5}$ and batch size 60, noise with probability $5 \times 10^{-1}$, AdamW optimizer | 0,43376 | 0,43246 |
| Back-propagation | DenseNet121 with learning rate $10^{-3}$ and batch size 60, AdamW optimizer | 0,43824 | 0,43989 |
| Back-propagation | DenseNet121 with learning rate $10^{-3}$ and batch size 60, RMSProp optimizer | 0,43531 | 0,43592 |
| Back-propagation | DenseNet121 with learning rate $10^{-3}$ and batch size 60, SGD optimizer | 0,01091 | 0,01105 |
| Back-propagation | DenseNet121 with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,43808 | 0,43807 |
| Back-propagation | DenseNet121 with learning rate $10^{-4}$ with linear decay and batch size 60, AdamW optimizer | 0,43549 | 0,43615 |
| Back-propagation | DenseNet121 with learning rate $10^{-4}$ with linear decay and batch size 60, noise with probability $5 \times 10^{-1}$, AdamW optimizer | 0,43437 | 0,43484 |
| Back-propagation | DenseNet121 with learning rate $10^{-5}$ and batch size 60, 2-layered head with 4187 hidden nodes, AdamW optimizer | 0,43247 | 0,43293 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-3}$ and batch size 60, AdamW optimizer | 0,43690 | 0,43715 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-3}$ and batch size 60, BCE loss, AdamW optimizer | 0,40232 | 0,40286 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-3}$ and batch size 60, cyclic lr from $10^{-5}$ to $10^{-1}$ with annealing ns=4-16, AdamW optimizer | 0,41965 | 0,41725 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-3}$ and batch size 60, cyclic lr from $10^{-5}$ to $10^{-1}$ with annealing ns=4-24, AdamW optimizer | 0,42813 | 0,42777 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-3}$ with linear decay and batch size 60, AdamW optimizer | 0,43727 | 0,43903 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-4}$ and batch size 60, AdamW optimizer | 0,43891 | 0,43932 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-5}$ and batch size 60, AdamW optimizer | 0,43463 | 0,43526 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-5}$ with linear decay and batch size 60, AdamW optimizer | 0,43149 | 0,43231 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-3}$ with linear decay and batch size 60, AdamW optimizer | 0,43727 | 0,43903 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-3}$ and batch size 60, 2-layered head with 4187 hidden nodes, AdamW optimizer | 0,20572 | 0,20604 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-4}$ and batch size 60, AdamW optimizer | 0,43891 | 0,43932 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-5}$ and batch size 60, AdamW optimizer | 0,43463 | 0,43526 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-5}$ with linear decay and batch size 60, AdamW optimizer | 0,43149 | 0,43231 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-5}$ and batch size 60, 2-layered head with 16748 hidden nodes, AdamW optimizer | 0,43630 | 0,43786 |
| Back-propagation | DenseNet121 with learning rate $5 \times 10^{-5}$ and batch size 60, 2-layered head with 4187 hidden nodes, AdamW optimizer | 0,43478 | 0,43563 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$ and batch size 60, 2-layered head with 4187 hidden nodes, AdamW optimizer | 0,43839 | 0,43858 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$ and batch size 60, dropout with probability $10^{-1}$, AdamW optimizer | 0,07096 | 0,07154 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$ and batch size 60, dropout with probability $2 \times 10^{-1}$, AdamW optimizer | 0,04754 | 0,04780 |
| Back-propagation | DenseNet161 with learning rate $10^{-5}$ and batch size 60, AdamW optimizer | 0,43214 | 0,43309 |
| Back-propagation | DenseNet161 with learning rate $5 \times 10^{-4}$ | 0,44441 | 0,44494 |
| Back-propagation | DenseNet161 with learning rate $5 \times 10^{-4}$, AdamW optimizer | 0,44429 | 0,44517 |
| Back-propagation | DenseNet161 with learning rate $5 \times 10^{-4}$, softmax activation, AdamW optimizer | 0,31181 | 0,31187 |
| Back-propagation | DenseNet161 with learning rate $5 \times 10^{-4}$ and BCE loss | 0,39090 | 0,39252 |
| Back-propagation | DenseNet161 with learning rate $5 \times 10^{-4}$, cyclic lr from $10^{-5}$ to $10^{-1}$ with ns=4, AdamW optimizer | 0,41467 | 0,41618 |
| Back-propagation | DenseNet161 with learning rate $5 \times 10^{-4}$ without gradient clipping.pth | 0,44430 | 0,44524 |
| Back-propagation | DenseNet161 with learning rate $5 \times 10^{-4}$, noise with probabiliy $5 \times 10^{-2}$, AdamW optimizer | 0,44070 | 0,44114 |
| Back-propagation | DenseNet161 with learning rate $5 \times 10^{-5}$ and batch size 60, AdamW optimizer | 0,44181 | 0,44230 |
| Back-propagation | DenseNet161 with learning rate $10^{-2}$ with linear decay, AdamW optimizer | 0,43979 | 0,44167 |
| Back-propagation | DenseNet161 with learning rate $10^{-3}$ | 0,44461 | 0,44614 |
| Back-propagation | DenseNet161 with learning rate $10^{-3}$ with linear decay, AdamW optimizer | 0,44446 | 0,44505 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$ | 0,44216 | 0,44287 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$, 50 epochs, AdamW optimizer | 0,44328 | 0,44434 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,44311 | 0,44373 |
| Back-propagation | DenseNet161 with learning rate $5 \times 10^{-3}$ and batch size 60, cyclic lr from $10^{-5}$ to $10^{-1}$ with annealing ns=4-24, AdamW optimizer | 0,42813 | 0,42777 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$ and batch size 60, 2-layered head with 16748 hidden nodes, AdamW optimizer | 0,43968 | 0,44126 |
| Back-propagation | DenseNet161 with learning rate $10^{-2}$ with linear decay, AdamW optimizer | 0,43979 | 0,44167 |
| Back-propagation | DenseNet161 with learning rate $10^{-3}$ | 0,44461 | 0,44614 |
| Back-propagation | DenseNet161 with learning rate $10^{-3}$ with linear decay, AdamW optimizer | 0,44446 | 0,44505 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$ | 0,44216 | 0,44287 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$, 50 epochs, AdamW optimizer | 0,44328 | 0,44434 |
| Back-propagation | DenseNet161 with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,44311 | 0,44373 |
| Back-propagation | DenseNet161 trained for 80 and finetuned for 30 epochs, cyclic lr from $10^{-5}$ to $10^{-1}$ with ns=4, AdamW optimizer | 0,30641 | 0,30654 |
| Back-propagation | DenseNet161 trained for 80 and finetuned for 30 epochs, cyclic lr from $10^{-5}$ to $10^{-1}$ with ns=4, AdamW optimizer | 0,32418 | 0,32654 |
| Back-propagation | ResNet50 with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,42419 | 0,42421 |
| Back-propagation | ResNet50 with learning rate $10^{-5}$ and batch size 60, AdamW optimizer | 0,41061 | 0,41103 |
| Back-propagation | ResNet50 with learning rate $5 \times 10^{-5}$ and batch size 60, AdamW optimizer | 0,42258 | 0,42249 |
| Back-propagation | ResNet50 with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,42419 | 0,42421 |
| Back-propagation | ResNet50 with learning rate $10^{-5}$ and batch size 60, AdamW optimizer | 0,41061 | 0,41103 |
| Back-propagation | ResNet50 with learning rate $5 \times 10^{-5}$ and batch size 60, AdamW optimizer | 0,42258 | 0,42249 |
| Back-propagation | ResNet101 with learning rate $10^{-5}$ and batch size 60, AdamW optimizer | 0,37006 | 0,36971 |
| Back-propagation | ResNet101 with learning rate $5 \times 10^{-5}$ and batch size 60, AdamW optimizer | 0,42586 | 0,42539 |
| Back-propagation | ResNet101 with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,42787 | 0,42715 |
| Back-propagation | ResNet101 with learning rate $5 \times 10^{-4}$ and batch size 60, AdamW optimizer | 0,42588 | 0,42609 |
| Back-propagation | VGG-13 with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,43584 | 0,43658 |
| Back-propagation | VGG-13 with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,43584 | 0,43658 |
| Back-propagation | VGG-13 with learning rate $5 \times 10^{-5}$ and batch size 60, AdamW optimizer | 0,43360 | 0,43492 |
| Back-propagation | VGG-16 with learning rate $10^{-4}$ and batch size 60, AdamW optimizer | 0,43760 | 0,43712 |
| Back-propagation | VGG-16 with learning rate $5 \times 10^{-5}$ and batch size 60, AdamW optimizer | 0,43297 | 0,43182 |
| Heuristic-based | 1-NN baseline with ResNet101 encoder | 0,11075 | 0,10860 |
| Heuristic-based | 10-NN baseline with ResNet101 encoder | 0,06254 | 0,06291 |
| Heuristic-based | 15-NN baseline with ResNet101 encoder | 0,05251 | 0,05282 |
| Heuristic-based | 1-NN baseline with ResNet50 encoder | 0,08117 | 0,07846 |
| Heuristic-based | 5-NN baseline with ResNet50 encoder | 0,05868 | 0,05722 |
| Heuristic-based | 10-NN baseline with ResNet50 encoder | 0,04283 | 0,04246 |
| Heuristic-based | 1-NN baseline with VGG-11 encoder | 0,11893 | 0,11988 |
| Heuristic-based | 10-NN baseline with VGG-11 encoder | 0,06252 | 0,06378 |
| Heuristic-based | 15-NN baseline with VGG-11 encoder | 0,04994 | 0,05044 |
| Heuristic-based | 1-NN baseline with VGG-16 encoder | 0,25203 | 0,25277 |
| Heuristic-based | 10-NN baseline with VGG-16 encoder | 0,09261 | 0,09315 |
| Heuristic-based | 15-NN baseline with VGG-16 encoder | 0,07128 | 0,07202 |
| Heuristic-based | 1-NN baseline with DenseNet121 encoder | 0,08607 | 0,09207 |
| Heuristic-based | 10-NN baseline with DenseNet121 encoder | 0,05488 | 0,05610 |
| Heuristic-based | 15-NN baseline with DenseNet121 encoder | 0,04449 | 0,04511 |
| Heuristic-based | 1-NN baseline with DenseNet161 encoder | 0,07145 | 0,07246 |
| Heuristic-based | 10-NN baseline with DenseNet161 encoder | 0,05361 | 0,05483 |
| Heuristic-based | 15-NN baseline with DenseNet161 encoder | 0,04357 | 0,04459 |
| Heuristic-based | 1-NN baseline with fully finetuned DenseNet161 encoder | 0,05062 | 0,05084 |
| Ensemble/Majority voting | Top-10 performing DenseNets ensemble from section 4.1.1(d), table 4.1.1 | 0,44545 | 0,44553 |
| Ensemble/Majority voting | CNNs ensemble from section 4.1.1(e), table 4.1.2 | 0,44170 | 0,44167 |
| Ensemble/Majority voting | CNNs ensemble from section 4.1.1(e), table 4.1.3 | 0,44305 | 0,44380 |
| Ensemble/Majority voting | CNNs ensemble from section 4.1.1(e), table 4.1.4 | 0,44543 | 0,44624 |

Table 7.2: Summary of all IU-XRAY Concept Prediction experiments configurations' training targets and F1 scores using 90% for training and the remaining 10% as a hold-out validation set

| BACKBONE | TRAINING REGIME | DEEP LEARNING MODEL | VALIDATION F1 LOSS | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Average |
| ResNet50 | Back-propagation | ResNet 50 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,377853 | 0,332436 | 0,395627 | 0,329744 | 0,362569 | 0,359646 |
| ResNet50 | Back-propagation | ResNet 50 with learning rate 0.001 and batch size 300, Adam optimizer | 0,280360 | 0,255886 | 0,268447 | 0,329744 | 0,277444 | 0,282376 |
| ResNet50 | Back-propagation | ResNet 50 with learning rate 0.001 and batch size 400, Adam optimizer | 0,276979 | 0,257445 | 0,270429 | 0,329744 | 0,278048 | 0,282529 |
| ResNet50 | Back-propagation | ResNet 50 with learning rate 0.0005 and batch size 400, Adam optimizer | 0,203687 | 0,332436 | 0,396106 | 0,329744 | 0,366897 | 0,325774 |
| ResNet101 | Back-propagation | ResNet 101 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,386209 | 0,342066 | 0,397021 | 0,329744 | 0,363312 | 0,363671 |
| ResNet101 | Back-propagation | ResNet 101 with learning rate 0.0005 and batch size 400, Adam optimizer | 0,295151 | 0,340869 | 0,292163 | 0,329744 | 0,364465 | 0,324478 |
| ResNet101 | Back-propagation | ResNet 101 with learning rate 0.001 and batch size 400, Adam optimizer | 0,297303 | 0,270049 | 0,267873 | 0,329744 | 0,279759 | 0,288946 |
| ResNet101 | Back-propagation | ResNet 101 with learning rate 0.0001 and batch size 300, Adam optimizer | 0,373379 | 0,332436 | 0,394392 | 0,329744 | 0,358008 | 0,357592 |
| ResNet101 | Back-propagation | ResNet 101 with learning rate 0.0001 and batch size 400, Adam optimizer | 0,373379 | 0,332436 | 0,394347 | 0,329744 | 0,358008 | 0,357583 |
| DenseNet121 | Back-propagation | DenseNet121 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,412437 | 0,370498 | 0,423721 | 0,338978 | 0,398268 | 0,388780 |
| DenseNet161 | Back-propagation | DenseNet161 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,414259 | 0,377403 | 0,423390 | 0,337359 | 0,391498 | 0,388782 |
| VGG-13 | Back-propagation | VGG-13 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,393630 | 0,344903 | 0,409759 | 0,329296 | 0,377688 | 0,371055 |
| VGG-16 | Back-propagation | VGG-16 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,381985 | 0,351673 | 0,416327 | 0,330945 | 0,385988 | 0,373384 |

Table 7.3: Summary of all IU-XRAY Concept Prediction experiments configurations' training targets and F1 scores using 80% for training and the remaining 20% as a hold-out validation set

| BACKBONE | TRAINING REGIME | DEEP LEARNING MODEL | VALIDATION F1 LOSS | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Average |
| ResNet50 | Back-propagation | ResNet 50 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,363401 | 0,349356 | 0,305460 | 0,348587 | 0,358154 | 0,344992 |
| ResNet50 | Back-propagation | ResNet 50 with learning rate 0.002 and batch size 300, Adam optimizer | 0,270173 | 0,263681 | 0,261192 | 0,348587 | 0,269580 | 0,282643 |
| ResNet50 | Back-propagation | ResNet 50 with learning rate 0.001 and batch size 400, Adam optimizer | 0,269161 | 0,260835 | 0,267645 | 0,348587 | 0,265743 | 0,282394 |
| ResNet101 | Back-propagation | ResNet 101 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,361373 | 0,345222 | 0,379884 | 0,348587 | 0,360839 | 0,359181 |
| DenseNet | Back-propagation | DenseNet121 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,396165 | 0,370709 | 0,400684 | 0,355657 | 0,386378 | 0,381919 |
| DenseNet | Back-propagation | DenseNet161 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,401418 | 0,380476 | 0,402166 | 0,354166 | 0,383900 | 0,384425 |
| VGG-13 | Back-propagation | VGG-13 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,385224 | 0,367254 | 0,394278 | 0,348218 | 0,373469 | 0,373689 |
| VGG-16 | Back-propagation | VGG-16 with learning rate 0.0005 and batch size 300, Adam optimizer | 0,377928 | 0,358715 | 0,397625 | 0,349395 | 0,377372 | 0,372207 |

Table 7.4: P-Values for different types of T-test between different backbone network architectures. We observe that DenseNet is the best performing image encoder for concept detection, as it is also indicated by the validation F1 scores, while the performance difference is statistically significant compared to AlexNet and ResNet image encoders but we cannot reject the null hypothesis when comparing to VGG. Within each model architecture we incorporate instances of the backbones mentioned in chapter 4 and are illustrated in the summary view in Figure 4.1.3 (p. 33).

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **AlexNet vs DenseNet** | **F1 score** | 0,030264494 | 0,024690244 | 0,021105685 | 0,060528988 | 0,049380487 | 0,042211371 |
| **AlexNet vs ResNet** | **F1 score** | 0,164555888 | 0,108784503 | 0,107343013 | 0,329111776 | 0,217569006 | 0,214686026 |
| **AlexNet vs VGG** | **F1 score** | 0,053273041 | 0,032687292 | 0,004797858 | 0,106546083 | 0,065374585 | 0,009595715 |
| **DenseNet vs ResNet** | **F1 score** | 0,001030623 | 0,000597724 | 0,000248738 | 0,002061246 | 0,001195448 | 0,000497475 |
| **DenseNet vs VGG** | **F1 score** | 0,091228429 | 0,388635052 | 0,205509876 | 0,182456857 | 0,777270104 | 0,411019753 |
| **ResNet vs VGG** | **F1 score** | 0,004251937 | 0,000822348 | 4,13E-05 | 0,008503874 | 0,001644697 | 8,25E-05 |

Table 7.5: P-Values for different types of T-test between different training regimes. We compare architectures where we use pre-trained backbone networks on ImageNet classification dataset with baselines where we include the same image encoders combined with a heuristic approach based on 1-NN. In that case we observe that the fine-tuned classification heads' performance, which are initialized using Glorot formula as explained in previous parts, is better and the F1 difference is statistically significant for any network architecture.

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **DenseNet** | **F1 score** | 1,27E-21 | 5,08E-88 | 3,97E-35 | 2,54E-21 | 1,02E-87 | 7,93E-35 |
| **ResNet** | **F1 score** | 1,33E-07 | 9,29E-24 | 9,33E-09 | 2,65E-07 | 1,86E-23 | 1,87E-08 |
| **VGG** | **F1 score** | 5,65E-05 | 5,26E-10 | 6,04E-05 | 0,000113079 | 1,05E-09 | 0,000120898 |
| **All Networks** | **F1 score** | 4,76E-22 | 1,00E-123 | 8,20E-41 | 9,52E-22 | 2,00E-123 | 1,64E-40 |

Table 7.6: P-Values for different types of T-test between varying training extent. We compare baselines where we fully finetune the CNNs end-to-end using the same objective and cyclical learning rates with architectures where we use pre-trained backbone networks on ImageNet classification dataset andbaselines where we include the same image encoders combined with a heuristic approach based on 1-NN. Again we observe that the fine-tuned classification heads' performance, which are initialized using Glorot formula as explained previously, yields higher scores than the fully finetuned visuin encoders and both are better than the 1-NN baseline The F1 score differences in both cases is statistically significant for any network architecture (either DenseNet121 or DenseNet161).

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **FFT vs CLS Head** | **F1 score** | 0,000151978 | 1,37E-18 | 1,59E-05 | 0,000303956 | 2,74E-18 | 3,19E-05 |
| **FFT vs NN Baselines** | **F1 score** | 8,76E-05 | 1,09E-17 | 6,56E-09 | 0,000175297 | 2,19E-17 | 1,31E-08 |

Table 7.7: P-Values for different types of T-test when incorporating mixtures of experts to take advantage of the *wizdom of the crowd*. We compare architectures where we use pre-trained backbone networks on ImageNet classification dataset with their ensembles. In that case we observe that the fine-tuned classification heads' performance, which are initialized using Glorot formula as explained in previous parts, perform better when we include the same image encoders and the F1 difference is statistically significant but we cannot reject the null hypothesis when comparing to the top-10 best performing DenseNets ensemble. In the latter case we take into consideration only DenseNet weak learners for a fair performance comparison; yet we still fail to reject the null hypothesis.

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **DenseNet** | **F1 score** | 0,05453075 | 0,212824857 | 2,78E-07 | 0,109061499 | 0,425649714 | 5,56E-07 |
| **All Networks** | **F1 score** | 0,008963781 | 0,036620236 | 3,62E-12 | 0,017927562 | 0,073240473 | 7,24E-12 |

Table 7.8: P-Values and Cohen's d for different types of statistical tests between different backbone network architectures. We observe a high variance in the results related to DenseNet compared to AlexNet and ResNet image encoders, where DenseNet is the best performing image encoder, other than that Wilcoxon signed-rank tests and Mann-Whitney U tests agree the performance difference is statistically significant. Within each model architecture we incorporate instances of the backbones mentioned in chapter 4 and are illustrated in the summary view in Figure 4.1.3 (p. 33).

| | *Metrics* | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **AlexNet vs DenseNet** | **F1 score** | 0,62865797 | 2,95E-06 | 5,90E-06 | -0,472637742 | 2,90E-06 | 5,81E-06 | 5,08E-07 |
| **AlexNet vs ResNet** | **F1 score** | 0,566022261 | 0,001417584 | 0,002835167 | 0,387048017 | 0,001363554 | 0,002727107 | 0,000180779 |
| **AlexNet vs VGG** | **F1 score** | 5,71E-08 | 0,001070028 | 0,002140056 | -0,729477003 | 0,001001324 | 0,002002648 | 0,017013643 |
| **DenseNet vs ResNet** | **F1 score** | 0,254962329 | 7,84E-09 | 1,57E-08 | 0,82278301 | 7,68E-09 | 1,54E-08 | 6,83E-12 |
| **DenseNet vs VGG** | **F1 score** | 1,47E-08 | 0,0379253 | 0,075850601 | -0,094553007 | 0,037464111 | 0,074928222 | 0,003324937 |
| **ResNet vs VGG** | **F1 score** | 1,97E-07 | 5,90E-06 | 1,18E-05 | -1,349230466 | 5,43E-06 | 1,09E-05 | 5,73E-07 |

Table 7.9: P-Values and Cohen's d for different types of statistical tests between different training regimes. We again observe a high variance in the results related to DenseNet and ResNet image encoders when comparing architectures where we use pre-trained backbone networks on ImageNet classification dataset with baselines where we include the same image encoders combined with a heuristic approach based on 1-NN. Yet Wilcoxon signed-rank tests and Mann-Whitney U tests agree that the fine-tuned classification heads' perform better and the F1 difference is statistically significant for any network architecture.

| | *Metrics* | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **DenseNet** | **F1 score** | 0,118516543 | 1,08E-10 | 2,15E-10 | 18,32153013 | 1,05E-10 | 2,10E-10 | 3,33E-13 |
| **ResNet** | **F1 score** | 0,22368679 | 0,000144607 | 0,000289214 | 18,57681715 | 0,000130365 | 0,00026073 | 4,24E-05 |
| **VGG** | **F1 score** | 1,40E-12 | 0,000676627 | 0,001353254 | 7,331440282 | 0,000569048 | 0,001138096 | 0,000264178 |
| **All Networks** | **F1 score** | 8,18E-06 | 0,00E+00 | 0,00E+00 | 10,80255796 | 1,85E-22 | 3,69E-22 | 0 |

Table 7.10: P-Values and Cohen's d for different types of statistical tests between varying training extent. We again observe a high variance in the results when we compare baselines where we fully finetune the CNNs end-to-end using the same objective and cyclical learning rates with architectures where we use pre-trained backbone networks on ImageNet classification dataset and baselines where we include the same image encoders combined with a heuristic approach based on the 1-NN but Wilcoxon signed-rank tests and Mann-Whitney U tests agree that the fine-tuned classification heads' performanceyields higher scores than the fully finetuned visuin encoders and both are better than the 1-NN baseline The F1 score differences in both cases is statistically significant for any network architecture (either DenseNet121 or DenseNet161).

| | *Metrics* | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **FFT vs CLS Head** | **F1 score** | 0,297403364 | 0,000505087 | 0,001010175 | 5,579063629 | 0,000488997 | 0,000977994 | 0,000332829 |
| **FFT vs NN Baselines** | **F1 score** | 0,675752856 | 0,00144732 | 0,002894639 | 18,04803036 | 0,001248454 | 0,002496909 | 0,000849059 |

Table 7.11: P-Values and Cohen's d for different types of statistical tests when incorporating mixtures of experts to take advantage of the wizdom of the crowd. In the case we compare architectures where we use pre-trained backbone networks on ImageNet classification dataset with their ensembles the variance is small and Wilcoxon signed-rank tests and Mann-Whitney U tests agree that the fine-tuned classification heads' performance perform better when we use the same image encoders and the F1 difference is statistically significant but we cannot reject the null hypothesis when comparing to the top-10 best performing DenseNets ensemble. In the latter case, the effect size is small, while we consider only DenseNet weak learners for a fair performance comparison; yet we still fail to reject the null hypothesis.

| | *Metrics* | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **DenseNet** | **F1 score** | 0,004683719 | 0,011028956 | 0,022057912 | -0,027967009 | 0,010651747 | 0,021303495 | 0,015208194 |
| **All Networks** | **F1 score** | 7,72E-08 | 4,73E-05 | 9,46E-05 | -0,041289257 | 4,65E-05 | 9,31E-05 | 0,000489134 |

---

Table 7.12: P-Values for different types of T-test when incorporating mixtures of experts to take advantage of the wizdom of the crowd. We compare our submissions where we use pre-trained backbone networks on ImageNet classification dataset with their ensembles. In that case we observe that the fine-tuned classification heads' performance, which are initialized using Glorot formula as explained in previous parts, perform better when we include the same image encoders and the F1 difference is statistically significant according to most tests.

| | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|
| | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **F1 score** | 0,032647976 | 0,021363965 | 0,040604872 | 0,065295952 | 0,04272793 | 0,081209744 |
| **Manual F1 score** | 0,014124286 | 0,000836787 | 0,000850966 | 0,028248572 | 0,001673573 | 0,001701933 |

Table 7.13: P-Values for different types of T-test between different training regimes. We compare our submissions where we use pre-trained backbone networks on ImageNet classification dataset with fully finetuned CNNs end-to-end using the same objective and cyclical learning rates with architectures where we use pre-trained backbone networks on ImageNet classification dataset andbaselines where we include the same image encoders combined with a heuristic approach based on 1-NN. In order to reject the null hypothesis we need to make the assumption of equal variance, which is possible though according to the F-test in table 5.3.13, which indicates a low variance in the F1 scores. Without making this assumption a priori we fail to reject the null hypothesis according to most T-tests among those performed.

| | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|
| | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **F1 score** | 0,067927343 | 0,000850571 | 0,068342062 | 0,135854686 | 0,001701142 | 0,136684125 |
| **Manual F1 score** | 0,101665515 | 0,00413529 | 0,103117507 | 0,203331029 | 0,008270581 | 0,206235015 |

Table 7.14: P-Values for different types of T-test between different training regimes. We compare our submissions where we use fully finetuned CNNs end-to-end using the same objective and cyclical learning rates with architectures where we use pre-trained backbone networks on ImageNet classification dataset and baselines where we include the same image encoders combined with a heuristic approach based on 1-NN to ensemble networks taking advantage of the wizdom of the crowd. In order to reject the null hypothesis we need to make the assumption of equal variance, which is possible though according to the F-test in table 5.3.14, which indicates a low variance in the F1 scores. Without making this assumption a priori we fail to reject the null hypothesis according to most tests shown.

| | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|
| | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **F1 score** | 0,067928104 | 0,000936068 | 0,069551811 | 0,135856208 | 0,001872136 | 0,139103622 |
| **Manual F1 score** | 0,102984509 | 0,003806526 | 0,10087491 | 0,205969017 | 0,007613052 | 0,201749819 |

Table 7.15: P-Values and Cohen's d for different types of statistical tests when incorporating mixtures of experts to take advantage of the wisdom of the crowd. We compare our submissions where we use pre-trained backbone networks on ImageNet classification dataset with their ensembles. In that case we observe a high variance in the manual F1 score, however Wilcoxon signed-rank tests and Mann-Whitney U tests agree the performance difference is statistically significant, thus the fine-tuned classification heads perform better than other baselines and the F1 difference is statistically significant according to most tests.

| | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|
| | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **F1 score** | 0,003812565 | 0,015191411 | 0,030382822 | 1,812410457 | 0,010460668 | 0,020921335 | 0,011065637 |
| **Manual F1 score** | 0,907534797 | 0,015191411 | 0,030382822 | -3,814032034 | 0,010460668 | 0,020921335 | 0,011065637 |

Table 7.16: P-Values and Cohen's d for different types of statistical tests between different training regimes. We compare our submissions where we use pre-trained backbone networks on ImageNet classification dataset with fully finetuned CNNs end-to-end using the same objective and cyclical learning rates with architectures where we use pre-trained backbone networks on ImageNet classification dataset and baselines where we include the same image encoders combined with a heuristic approach based on 1-NN. In that case, we fail to reject the null hypothesis according to the Wilcoxon signed-rank tests and Mann-Whitney U tests, despite the low variance indicated by the F-test.

| | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|
| | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **F1 score** | 7,48E-07 | 0,052596253 | 0,105192505 | 6,484786644 | 0,032038753 | 0,064077506 | 0,046865909 |
| **Manual F1 score** | 8,96E-06 | 0,052596253 | 0,105192505 | 4,210144559 | 0,032038753 | 0,064077506 | 0,046865909 |

Table 7.17: P-Values and Cohen's d for different types of statistical tests between different training regimes. We compare our submissions where we use fully finetuned CNNs end-to-end using the same objective and cyclical learning rates with architectures where we use pre-trained backbone networks on ImageNet classification dataset and baselines where we include the same image encoders combined with a heuristic approach based on 1-NN to ensemble networks taking advantage of the wizdom of the crowd. In that case, we fail to reject the null hypothesis according to the Wilcoxon signed-rank tests and Mann-Whitney U tests, despite the low variance indicated by the F-test.

| | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|
| | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **F1 score** | 0,000647182 | 0,052596253 | 0,105192505 | 6,322196713 | 0,032038753 | 0,064077506 | 0,046865909 |
| **Manual F1 score** | 1,11E-05 | 0,052596253 | 0,105192505 | 4,310285807 | 0,032038753 | 0,064077506 | 0,046865909 |

Table 7.18: P-Values for different types of T-test between WinnerTakesAll and 1-NN baselines

|  | *Metrics* | **One tailed *T*-test (default)** | **One tailed *T*-test equal variance** | **One tailed *T*-test unequal variance** | **Two tailed *T*-test (default)** | **Two tailed *T*-test equal variance** | **Two tailed *T*-test unequal variance** |
|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,002438605 | 0,001358262 | 0,002438605 | 0,004877209 | 0,002716524 | 0,004877209 |
| **BLUE2** | Precision | 4,58E-06 | 2,17E-07 | 4,58E-06 | 9,16E-06 | 4,35E-07 | 9,16E-06 |
| **BLUE3** |  | 6,48E-05 | 1,09E-05 | 6,48E-05 | 0,000129673 | 2,18E-05 | 0,000129673 |
| **BLUE4** |  | 0,000948265 | 0,000411969 | 0,000948265 | 0,001896529 | 0,000823938 | 0,001896529 |
| **ROUGE_L** | Recall | 0,000505046 | 0,000181297 | 0,000505046 | 0,001010091 | 0,000362594 | 0,001010091 |
| **METEOR** |  | 2,53E-12 | 2,46E-18 | 2,53E-12 | 5,06E-12 | 4,91E-18 | 5,06E-12 |
| **CIDER** |  | 0,000869733 | 0,000368521 | 0,000869733 | 0,001739467 | 0,000737042 | 0,001739467 |

Table 7.19: P-Values for different types of T-test between WinnerTakesAll and 1-NN with Retrieval Augmented Generation baselines

|  | *Metrics* | **One tailed *T*-test (default)** | **One tailed *T*-test equal variance** | **One tailed *T*-test unequal variance** | **Two tailed *T*-test (default)** | **Two tailed *T*-test equal variance** | **Two tailed *T*-test unequal variance** |
|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,001615772 | 1,20E-42 | 0,001615772 | 0,003231545 | 2,40E-42 | 0,003231545 |
| **BLUE2** | Precision | 0,007668634 | 1,79E-30 | 0,007668634 | 0,015337267 | 3,57E-30 | 0,015337267 |
| **BLUE3** |  | 0,011783871 | 4,06E-27 | 0,011783871 | 0,023567742 | 8,13E-27 | 0,023567742 |
| **BLUE4** |  | 0,02118931 | 1,55E-22 | 0,02118931 | 0,042378621 | 3,11E-22 | 0,042378621 |
| **ROUGE_L** | Recall | 0,00E+00 | 8,85E-272 | 2,88E-265 | 0,00E+00 | 1,77E-271 | 5,77E-265 |
| **METEOR** |  | 0,009642803 | 1,10E-28 | 0,009642803 | 0,019285607 | 2,20E-28 | 0,019285607 |
| **CIDER** |  | 0,00487171 | 5,08E-34 | 0,00487171 | 0,00974342 | 1,02E-33 | 0,00974342 |

Table 7.20: P-Values for different types of T-test between WinnerTakesAll baseline and Pegasus abstractive summarization algorithm

|  | *Metrics* | **One tailed *T*-test (default)** | **One tailed *T*-test equal variance** | **One tailed *T*-test unequal variance** | **Two tailed *T*-test (default)** | **Two tailed *T*-test equal variance** | **Two tailed *T*-test unequal variance** |
|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,007396865 | 0,002242038 | 0,007396865 | 0,014793729 | 0,004484076 | 0,014793729 |
| **BLUE2** | Precision | 7,15E-06 | 6,40E-09 | 7,15E-06 | 1,43E-05 | 1,28E-08 | 1,43E-05 |
| **BLUE3** |  | 3,53E-06 | 1,59E-09 | 3,53E-06 | 7,05E-06 | 3,18E-09 | 7,05E-06 |
| **BLUE4** |  | 3,99E-06 | 2,03E-09 | 3,99E-06 | 7,98E-06 | 4,06E-09 | 7,98E-06 |
| **ROUGE_L** | Recall | 2,58E-05 | 7,91E-08 | 2,58E-05 | 5,17E-05 | 1,58E-07 | 5,17E-05 |
| **METEOR** |  | 1,35E-05 | 2,23E-08 | 1,35E-05 | 2,70E-05 | 4,45E-08 | 2,70E-05 |
| **CIDER** |  | 1,39E-06 | 2,52E-10 | 1,39E-06 | 2,78E-06 | 5,04E-10 | 2,78E-06 |

Table 7.21: P-Values for different types of T-test between WinnerTakesAll and 1-NN with Pegasus abstractive summarization baselines

|  | *Metrics* | **One tailed *T*-test (default)** | **One tailed *T*-test equal variance** | **One tailed *T*-test unequal variance** | **Two tailed *T*-test (default)** | **Two tailed *T*-test equal variance** | **Two tailed *T*-test unequal variance** |
|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 3,69E-07 | 1,34E-10 | 3,69E-07 | 7,39E-07 | 2,68E-10 | 7,39E-07 |
| **BLUE2** | Precision | 4,93E-09 | 2,94E-14 | 4,93E-09 | 9,87E-09 | 5,88E-14 | 9,87E-09 |
| **BLUE3** |  | 6,34E-09 | 4,82E-14 | 6,34E-09 | 1,27E-08 | 9,63E-14 | 1,27E-08 |
| **BLUE4** |  | 1,80E-08 | 3,75E-13 | 1,80E-08 | 3,60E-08 | 7,49E-13 | 3,60E-08 |
| **ROUGE_L** | Recall | 4,66E-12 | 2,86E-20 | 4,66E-12 | 9,31E-12 | 5,72E-20 | 9,31E-12 |
| **METEOR** |  | 1,97E-07 | 3,98E-11 | 1,97E-07 | 3,93E-07 | 7,95E-11 | 3,93E-07 |
| **CIDER** |  | 2,87E-06 | 6,61E-09 | 2,87E-06 | 5,74E-06 | 1,32E-08 | 5,74E-06 |

Table 7.22: P-Values for different types of T-test between 1-NN (simple) and 1-NN with Retrieval Augmented Generation baselines

|  | *Metrics* | **One tailed *T*-test (default)** | **One tailed *T*-test equal variance** | **One tailed *T*-test unequal variance** | **Two tailed *T*-test (default)** | **Two tailed *T*-test equal variance** | **Two tailed *T*-test unequal variance** |
|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 2,18E-10 | 0,004513418 | 3,47E-08 | 4,37E-10 | 0,009026836 | 6,94E-08 |
| **BLUE2** | Precision | 0,00E+00 | 0,001567471 | 2,19E-08 | 0,00E+00 | 0,003134941 | 4,38E-08 |
| **BLUE3** |  | 0,018702559 | 0,00060651 | 1,36E-06 | 0,037405118 | 0,001213019 | 2,72E-06 |
| **BLUE4** |  | 0,00E+00 | 0,000702426 | 0,000906843 | 0,00E+00 | 0,001404851 | 0,001813685 |
| **ROUGE_L** | Recall | 0,009092095 | 0,001507451 | 3,62E-09 | 0,018184189 | 0,003014902 | 7,25E-09 |
| **METEOR** |  | 0,00E+00 | 0,000695965 | 6,57E-10 | 0,00E+00 | 0,001391929 | 1,31E-09 |
| **CIDER** |  | 0,021917701 | 0,000403516 | 2,55E-10 | 0,043835403 | 0,000807031 | 5,10E-10 |

Table 7.23: P-Values for different types of T-test between 1-NN (simple) and Pegasus abstractive summarization algorithm

|  | *Metrics* | **One tailed *T*-test (default)** | **One tailed *T*-test equal variance** | **One tailed *T*-test unequal variance** | **Two tailed *T*-test (default)** | **Two tailed *T*-test equal variance** | **Two tailed *T*-test unequal variance** |
|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,008791795 | 0,356893136 | 0,295077556 | 0,01758359 | 0,713786271 | 0,590155112 |
| **BLUE2** | Precision | 0,015001546 | 0,387155532 | 0,319198546 | 0,030003091 | 0,774311063 | 0,638397092 |
| **BLUE3** |  | 0,042763296 | 0,346695771 | 0,253216378 | 0,085526592 | 0,693391542 | 0,506432756 |
| **BLUE4** |  | 0,058449106 | 0,39484979 | 0,323834878 | 0,116898213 | 0,78969958 | 0,647669757 |
| **ROUGE_L** | Recall | 0,007550328 | 0,495915193 | 0,493118378 | 0,015100656 | 0,991830385 | 0,986236755 |
| **METEOR** |  | 0,083263874 | 0,118126832 | 0,066060677 | 0,166527748 | 0,236253664 | 0,132121353 |
| **CIDER** |  | 0,017225987 | 0,171774769 | 0,053910108 | 0,034451975 | 0,343549538 | 0,107820215 |

Table 7.24: P-Values for different types of T-test between 1-NN (simple) and 1-NN with Pegasus abstractive summarization baselines

|  | *Metrics* | **One tailed *T*-test (default)** | **One tailed *T*-test equal variance** | **One tailed *T*-test unequal variance** | **Two tailed *T*-test (default)** | **Two tailed *T*-test equal variance** | **Two tailed *T*-test unequal variance** |
|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,029341555 | 0,000194712 | 1,48E-05 | 0,05868311 | 0,000389423 | 2,96E-05 |
| **BLUE2** | Precision | 0,02122989 | 4,92E-05 | 1,68E-06 | 0,042459779 | 9,85E-05 | 3,37E-06 |
| **BLUE3** |  | 0,015272507 | 2,13E-05 | 5,83E-07 | 0,030545014 | 4,27E-05 | 1,17E-06 |
| **BLUE4** |  | 0,017592744 | 3,70E-05 | 1,13E-06 | 0,035185489 | 7,41E-05 | 2,27E-06 |
| **ROUGE_L** | Recall | 0,003040734 | 4,67E-06 | 1,55E-07 | 0,006081469 | 9,35E-06 | 3,10E-07 |
| **METEOR** |  | 0,000204704 | 8,62E-08 | 1,19E-09 | 0,000409408 | 1,72E-07 | 2,38E-09 |
| **CIDER** |  | 0,076019151 | 0,000583578 | 0,000145621 | 0,152038302 | 0,001167157 | 0,000291241 |

Table 7.25: P-Values and Cohen's d for different types of statistical tests between WinnerTakesAll and 1-NN baselines

|  | *Metrics* | ***F*-test** | **One tailed Mann-Whitney *U*-test** | **Two tailed Mann-Whitney *U*-test** | **Cohen's *d* - Baseline Effect Size test** | **One tailed Wilcoxon signed-rank test** | **Two tailed Wilcoxon signed-rank test** | **Kolmogorov-Smirnov test** |
|---|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **Cohen's *d*** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 8,99E-258 | 0,035196866 | 0,070393731 | -0,784358805 | 0,04377325 | 0,0875465 | 0,000284657 |
| **BLUE2** | Precision | 1,48E-269 | 2,33E-08 | 4,67E-08 | -1,510839411 | 1,48E-07 | 2,97E-07 | 4,43E-09 |
| **BLUE3** |  | 0,00E+00 | 2,33E-08 | 4,67E-08 | -1,193199833 | 1,48E-07 | 2,97E-07 | 4,43E-09 |
| **BLUE4** |  | 0,00E+00 | 5,92E-07 | 1,18E-06 | -0,89016265 | 2,61E-06 | 5,21E-06 | 3,81E-08 |
| **ROUGE_L** | Recall | 1,44E-250 | 0,001231793 | 0,002463585 | -0,960561058 | 0,002203385 | 0,00440677 | 1,16E-05 |
| **METEOR** |  | 6,88E-251 | 2,28E-08 | 4,56E-08 | 4,0745109 | 1,48E-07 | 2,97E-07 | 4,43E-09 |
| **CIDER** |  | 0,00E+00 | 2,31E-08 | 4,63E-08 | -0,899818691 | 1,48E-07 | 2,97E-07 | 4,43E-09 |

Table 7.26: P-Values and Cohen's d for different types of statistical tests between WinnerTakesAll and 1-NN with Retrieval Augmented Generation baselines

|  | *Metrics* | ***F*-test** | **One tailed Mann-Whitney *U*-test** | **Two tailed Mann-Whitney *U*-test** | **Cohen's *d* - Baseline Effect Size test** | **One tailed Wilcoxon signed-rank test** | **Two tailed Wilcoxon signed-rank test** | **Kolmogorov-Smirnov test** |
|---|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **Cohen's *d*** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 1,70E-223 | 1,16E-05 | 2,32E-05 | -197 | 0,011671101 | 0,023342202 | 0,017035766 |
| **BLUE2** | Precision | 4,38E-249 | 1,16E-05 | 2,32E-05 | -41,5 | 0,011671101 | 0,023342202 | 0,017035766 |
| **BLUE3** |  | 0,00E+00 | 1,16E-05 | 2,32E-05 | -26,999995 | 0,011671101 | 0,023342202 | 0,017035766 |
| **BLUE4** |  | 0,00E+00 | 1,16E-05 | 2,32E-05 | -14,999996 | 0,011671101 | 0,023342202 | 0,017035766 |
| **ROUGE_L** | Recall | 1,00E+00 | 1,13E-05 | 2,26E-05 | -4,11E+15 | 0,011671101 | 0,023342202 | 0,017035766 |
| **METEOR** |  | 1,70E-223 | 1,16E-05 | 2,32E-05 | 33 | 0,011671101 | 0,023342202 | 0,017035766 |
| **CIDER** |  | 0,00E+00 | 1,16E-05 | 2,32E-05 | -65,33333 | 0,011671101 | 0,023342202 | 0,017035766 |

Table 7.27: P-Values and Cohen's d for different types of statistical tests between WinnerTakesAll baseline and Pegasus abstractive summarization algorithm

|  | *Metrics* | ***F*-test** | **One tailed Mann-Whitney *U*-test** | **Two tailed Mann-Whitney *U*-test** | **Cohen's *d* - Baseline Effect Size test** | **One tailed Wilcoxon signed-rank test** | **Two tailed Wilcoxon signed-rank test** | **Kolmogorov-Smirnov test** |
|---|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **Cohen's *d*** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,00E+00 | 0,024668088 | 0,049336176 | -1,631014289 | 0,027331968 | 0,054663936 | 0,012238153 |
| **BLUE2** | Precision | 2,89E-78 | 0,001362266 | 0,002724532 | -7,444444444 | 0,001973876 | 0,003947752 | 0,001299744 |
| **BLUE3** |  | 3,18E-105 | 0,001362266 | 0,002724532 | -8,590398699 | 0,001973876 | 0,003947752 | 0,001299744 |
| **BLUE4** |  | 5,12E-104 | 0,001283339 | 0,002566678 | -8,378986159 | 0,001973876 | 0,003947752 | 0,001299744 |
| **ROUGE_L** | Recall | 0,00E+00 | 0,001389215 | 0,00277843 | -5,727155665 | 0,001973876 | 0,003947752 | 0,001299744 |
| **METEOR** |  | 3,19E-73 | 0,001362266 | 0,002724532 | 6,541684947 | 0,001973876 | 0,003947752 | 0,001299744 |
| **CIDER** |  | 1,83E-105 | 0,001362266 | 0,002724532 | -10,36857632 | 0,001973876 | 0,003947752 | 0,001299744 |

Table 7.28: P-Values and Cohen's d for different types of statistical tests between WinnerTakesAll and 1-NN with Pegasus abstractive summarization baselines

|  | *Metrics* | ***F*-test** | **One tailed Mann-Whitney *U*-test** | **Two tailed Mann-Whitney *U*-test** | **Cohen's *d* - Baseline Effect Size test** | **One tailed Wilcoxon signed-rank test** | **Two tailed Wilcoxon signed-rank test** | **Kolmogorov-Smirnov test** |
|---|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **Cohen's *d*** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,00E+00 | 7,91E-05 | 0,000158151 | -1,651909835 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **BLUE2** | Precision | 1,88E-126 | 7,91E-05 | 0,000158151 | -8,464657121 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **BLUE3** |  | 1,37E-170 | 7,77E-05 | 0,000155338 | -2,918611806 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **BLUE4** |  | 1,29E-169 | 7,63E-05 | 0,000152562 | -7,184156875 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **ROUGE_L** | Recall | 0,00E+00 | 7,70E-05 | 0,000153945 | -1,694654357 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **METEOR** |  | 2,11E-114 | 7,91E-05 | 0,000158151 | 5,289267563 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **CIDER** |  | 4,80E-177 | 8,05E-05 | 0,000161001 | -2,322267023 | 0,000174288 | 0,000348575 | 5,41E-05 |

Table 7.29: P-Values and Cohen's d for different types of statistical tests between 1-NN (simple) and Pegasus abstractive summarization algorithm

|  | *Metrics* | ***F*-test** | **One tailed Mann-Whitney *U*-test** | **Two tailed Mann-Whitney *U*-test** | **Cohen's *d* - Baseline Effect Size test** | **One tailed Wilcoxon signed-rank test** | **Two tailed Wilcoxon signed-rank test** | **Kolmogorov-Smirnov test** |
|---|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **Cohen's *d*** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,055797215 | 0,486692814 | 0,973385628 | 0,175147822 | 0,473423536 | 0,946847071 | 0,404165399 |
| **BLUE2** | Precision | 0,004510237 | 0,150408301 | 0,300816603 | -0,13684527 | 0,143061192 | 0,286122384 | 0,144200074 |
| **BLUE3** |  | 0,000944316 | 0,033110052 | 0,066220104 | -0,188316528 | 0,030974076 | 0,061948151 | 0,039375236 |
| **BLUE4** |  | 0,000239313 | 0,037805708 | 0,075611416 | -0,127265078 | 0,035930319 | 0,071860638 | 0,001171458 |
| **ROUGE_L** | Recall | 0,001259181 | 0,308269101 | 0,616538202 | -0,00488199 | 0,296901429 | 0,593802857 | 0,07784835 |
| **METEOR** |  | 0,2244522 | 0,070494617 | 0,140989234 | -0,574009395 | 0,066807201 | 0,133614403 | 0,039375236 |
| **CIDER** |  | 2,27E-06 | 0,5 | 1,00E+00 | 0,456335897 | 0,486704386 | 0,973408772 | 0,249995848 |

Table 7.30: P-Values and Cohen's d for different types of statistical tests between 1-NN (simple) and 1-NN with Pegasus abstractive summarization baselines

|  | *Metrics* | ***F*-test** | **One tailed Mann-Whitney *U*-test** | **Two tailed Mann-Whitney *U*-test** | **Cohen's *d* - Baseline Effect Size test** | **One tailed Wilcoxon signed-rank test** | **Two tailed Wilcoxon signed-rank test** | **Kolmogorov-Smirnov test** |
|---|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **Cohen's *d*** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,041513985 | 0,000842266 | 0,001684532 | -1,083080384 | 0,000780175 | 0,001560351 | 0,000512379 |
| **BLUE2** | Precision | 0,009997135 | 0,000704413 | 0,001408826 | -4,428001285 | 0,000653011 | 0,001306022 | 0,000512379 |
| **BLUE3** |  | 0,007863412 | 0,000702032 | 0,001404065 | -1,742946084 | 0,000653011 | 0,001306022 | 0,000512379 |
| **BLUE4** |  | 0,007610756 | 0,000753308 | 0,001506616 | -4,707917372 | 0,000713988 | 0,001427975 | 0,039028216 |
| **ROUGE_L** | Recall | 7,64E-06 | 2,63E-05 | 5,26E-05 | -1,131071149 | 2,42E-05 | 4,84E-05 | 4,08E-05 |
| **METEOR** |  | 0,001789036 | 1,65E-05 | 3,31E-05 | -8,645918132 | 1,55E-05 | 3,10E-05 | 2,31E-06 |
| **CIDER** |  | 0,188424684 | 0,001530465 | 0,003060931 | -1,407060364 | 0,001426266 | 0,002852533 | 0,000512379 |

Table 7.31: P-Values and Cohen's d for different types of statistical tests between 1-NN (simple) and 1-NN with Retrieval Augmented Generation baselines

|  | *Metrics* | ***F*-test** | **One tailed Mann-Whitney *U*-test** | **Two tailed Mann-Whitney *U*-test** | **Cohen's *d* - Baseline Effect Size test** | **One tailed Wilcoxon signed-rank test** | **Two tailed Wilcoxon signed-rank test** | **Kolmogorov-Smirnov test** |
|---|---|---|---|---|---|---|---|---|
|  |  | **P-Value** | **P-Value** | **P-Value** | **Cohen's *d*** | **P-Value** | **P-Value** | **P-Value** |
| **BLUE1** |  | 0,032526842 | 0,013705235 | 0,027410471 | -2,180791063 | 0,011671101 | 0,023342202 | 0,017035766 |
| **BLUE2** | Precision | 0,210367034 | 0,0135891 | 0,0271782 | -2,540228367 | 0,011671101 | 0,023342202 | 0,017035766 |
| **BLUE3** |  | 0,322171273 | 0,0135891 | 0,0271782 | -2,858432586 | 0,011671101 | 0,023342202 | 0,017035766 |
| **BLUE4** |  | 0,523716103 | 0,01558878 | 0,031177559 | -2,809325131 | 0,01373432 | 0,02746864 | 1,95E-06 |
| **ROUGE_L** | Recall | 0,00E+00 | 0,013560118 | 0,027120235 | -2,553362073 | 0,011671101 | 0,023342202 | 0,017035766 |
| **METEOR** |  | 0,082599613 | 0,013300208 | 0,026600415 | -2,812416198 | 0,011671101 | 0,023342202 | 0,017035766 |
| **CIDER** |  | 0,128696649 | 0,013473294 | 0,026946588 | -2,994756705 | 0,011671101 | 0,023342202 | 0,017035766 |

Table 7.32: P-Values for different types of T-test between 1-NN with Retrieval Augmented Generation baseline and Pegasus abstractive summarization algorithm

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **BLUE1** | Precision | 4,15E-11 | 0,000143049 | 1,68E-05 | 8,29E-11 | 0,000286099 | 3,36E-05 |
| **BLUE2** | | 0,009841499 | 6,24E-06 | 0,000102472 | 0,019682997 | 1,25E-05 | 0,000204945 |
| **BLUE3** | | 0,013441718 | 8,82E-07 | 0,006008318 | 0,026883436 | 1,76E-06 | 0,012016637 |
| **BLUE4** | | 0,023535527 | 1,03E-06 | 0,02416574 | 0,047071053 | 2,05E-06 | 0,048331481 |
| **ROUGE_L** | Recall | 0,00267481 | 7,97E-07 | 2,12E-07 | 0,00534962 | 1,59E-06 | 4,25E-07 |
| **METEOR** | | 0,006240571 | 0,001063701 | 9,50E-05 | 0,012481141 | 0,002127402 | 0,000189909 |
| **CIDER** | | 0,004694496 | 2,48E-10 | 0,002250419 | 0,009388991 | 4,96E-10 | 0,004500838 |

Table 7.33: P-Values for different types of T-test between 1-NN with Retrieval Augmented Generation and 1-NN with Pegasus abstractive summarization baselines

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **BLUE1** | Precision | 0,168573605 | 0,046277296 | 0,001562726 | 0,33714721 | 0,092554591 | 0,003125452 |
| **BLUE2** | | 0,105876136 | 0,007482975 | 0,001022543 | 0,211752272 | 0,01496595 | 0,002045086 |
| **BLUE3** | | 0,068781983 | 0,002112761 | 0,007010996 | 0,137563966 | 0,004225521 | 0,014021993 |
| **BLUE4** | | 0,050901928 | 0,002067128 | 0,04059562 | 0,101803856 | 0,004134255 | 0,08119124 |
| **ROUGE_L** | Recall | 0,08750652 | 0,000160645 | 8,09E-07 | 0,175013041 | 0,000321289 | 1,62E-06 |
| **METEOR** | | 0,42202087 | 0,122630125 | 0,014793618 | 0,844041739 | 0,24526025 | 0,029587235 |
| **CIDER** | | 0,101563007 | 0,004190111 | 2,85E-05 | 0,203126013 | 0,008380222 | 5,71E-05 |

Table 7.34: P-Values for different types of T-test between 1-NN with Pegasus abstractive summarization baseline and Dense Passage Retriever (based on the captions)

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **BLUE1** | Precision | 7,25E-08 | 5,72E-12 | 7,25E-08 | 1,45E-07 | 1,14E-11 | 1,45E-07 |
| **BLUE2** | | 2,68E-09 | 8,77E-15 | 2,68E-09 | 5,35E-09 | 1,75E-14 | 5,35E-09 |
| **BLUE3** | | 5,62E-10 | 3,99E-16 | 5,62E-10 | 1,12E-09 | 7,98E-16 | 1,12E-09 |
| **BLUE4** | | 4,27E-10 | 2,32E-16 | 4,27E-10 | 8,55E-10 | 4,63E-16 | 8,55E-10 |
| **ROUGE_L** | Recall | 1,02E-13 | 1,39E-23 | 1,02E-13 | 2,04E-13 | 2,79E-23 | 2,04E-13 |
| **METEOR** | | 8,21E-11 | 8,71E-18 | 8,21E-11 | 1,64E-10 | 1,74E-17 | 1,64E-10 |
| **CIDER** | | 3,21E-11 | 1,34E-18 | 3,21E-11 | 6,42E-11 | 2,68E-18 | 6,42E-11 |

Table 7.35: P-Values and Cohen's d for different types of statistical tests between 1-NN with Retrieval Augmented Generation baseline and Pegasus abstractive summarization algorithm

| | *Metrics* | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **BLUE1** | Precision | 0,077312591 | 0,033376508 | 0,066753015 | 6,141016785 | 0,022750132 | 0,045500264 | 0,032621652 |
| **BLUE2** | | 0,818073124 | 0,032566479 | 0,065132958 | 10,65481565 | 0,022750132 | 0,045500264 | 0,032621652 |
| **BLUE3** | | 0,553078009 | 0,032566479 | 0,065132958 | 14,87186097 | 0,022750132 | 0,045500264 | 0,032621652 |
| **BLUE4** | | 0,089770426 | 0,030149044 | 0,060298087 | 14,49568901 | 0,022750132 | 0,045500264 | 0,032621652 |
| **ROUGE_L** | Recall | 0,00E+00 | 0,032566479 | 0,065132958 | 15,12759048 | 0,022750132 | 0,045500264 | 0,032621652 |
| **METEOR** | | 0,138651168 | 0,032566479 | 0,065132958 | 4,200169686 | 0,022750132 | 0,045500264 | 0,032621652 |
| **CIDER** | | 0,324245623 | 0,032566479 | 0,065132958 | 58,53776476 | 0,022750132 | 0,045500264 | 0,032621652 |

Table 7.36: P-Values and Cohen's d for different types of statistical tests between 1-NN with Retrieval Augmented Generation and 1-NN with Pegasus abstractive summarization baselines

| | *Metrics* | *F*-test | Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **BLUE1** | Precision | 0,066417605 | 0,022076339 | 0,044152678 | 0,498477205 | 0,016947427 | 0,033894854 | 0,024271208 |
| **BLUE2** | | 0,52616189 | 0,022076339 | 0,044152678 | 2,344899001 | 0,016947427 | 0,033894854 | 0,024271208 |
| **BLUE3** | | 0,811085495 | 0,021591546 | 0,043183092 | 1,074397959 | 0,016947427 | 0,033894854 | 0,024271208 |
| **BLUE4** | | 0,771779403 | 0,02110919 | 0,04221838 | 2,980734475 | 0,016947427 | 0,033894854 | 0,024271208 |
| **ROUGE_L** | Recall | 0,00E+00 | 0,02110919 | 0,04221838 | 0,367044953 | 0,016947427 | 0,033894854 | 0,024271208 |
| **METEOR** | | 0,268189014 | 0,143321589 | 0,286643178 | 0,971729112 | 0,119296415 | 0,238592829 | 0,138631229 |
| **CIDER** | | 0,199980548 | 0,022563474 | 0,045126949 | 1,638842229 | 0,016947427 | 0,033894854 | 0,024271208 |

Table 7.37: P-Values and Cohen's d for different types of statistical tests between 1-NN with Pegasus abstractive summarization baseline and Dense Passage Retriever (based on the captions)

| | *Metrics* | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **BLUE1** | Precision | 0,00E+00 | 7,91E-05 | 0,000158151 | 2,037598034 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **BLUE2** | | 0,00E+00 | 7,91E-05 | 0,000158151 | 9,145888087 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **BLUE3** | | 3,36E-115 | 7,77E-05 | 0,000155338 | 3,962735908 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **BLUE4** | | 0,00E+00 | 7,63E-05 | 0,000152562 | 11,52522629 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **ROUGE_L** | Recall | 2,45E-109 | 7,70E-05 | 0,000153945 | 2,733313479 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **METEOR** | | 0,00E+00 | 7,91E-05 | 0,000158151 | 14,18221193 | 0,000174288 | 0,000348575 | 5,41E-05 |
| **CIDER** | | 1,97E-114 | 8,05E-05 | 0,000161001 | 9,947119477 | 0,000174288 | 0,000348575 | 5,41E-05 |

---

Table 7.38: P-Values for different types of T-test between 1-NN (simple) and 1-NN with Retrieval Augmented Generation submissions

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **BLUE_AVG** | Precision | 0,019430287 | 0,163003184 | 0,16368125 | 0,038860575 | 0,326006368 | 0,327362499 |
| **ROUGE_L** | Recall | 0,051078066 | 0,368008443 | 0,368137376 | 0,102156133 | 0,736016886 | 0,736274753 |
| **METEOR** | | 0,098355257 | 0,491171742 | 0,491171869 | 0,196710513 | 0,982343483 | 0,982343739 |
| **CIDER** | | 0,090620716 | 0,339411022 | 0,340219793 | 0,181241432 | 0,678822043 | 0,680439586 |
| **SPICE** | F1 | 0,107065258 | 0,292258288 | 0,295093571 | 0,214130517 | 0,584516577 | 0,590187143 |
| **BERTscore** | | 0,052045645 | 0,234558106 | 0,236246456 | 0,104091289 | 0,469116212 | 0,472492913 |

Table 7.39: P-Values for different types of T-test between 1-NN (simple) and Pegasus abstractive summarization algorithm submissions

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **BLUE_AVG** | Precision | 0,014417308 | 0,009516386 | 0,011154966 | 0,028834617 | 0,019032771 | 0,022309931 |
| **ROUGE_L** | Recall | 0,173717149 | 0,030110393 | 0,227425933 | 0,347434298 | 0,060220786 | 0,454851866 |
| **METEOR** | | 0,064453869 | 0,008480305 | 0,115626234 | 0,128907738 | 0,016960611 | 0,231252468 |
| **CIDER** | | 0,35693478 | 0,489638758 | 0,494540778 | 0,71386956 | 0,979277515 | 0,989081555 |
| **SPICE** | F1 | 0,356409242 | 0,458692453 | 0,480599205 | 0,712818484 | 0,917384906 | 0,961198409 |
| **BERTscore** | | 0,387060574 | 0,47245327 | 0,487281564 | 0,774121149 | 0,94490654 | 0,974563129 |

Table 7.40: P-Values for different types of T-test between 1-NN with Retrieval Augmented Generation and 1-NN with Pegasus abstractive summarization submissions

| | *Metrics* | One tailed *T*-test (default) | One tailed *T*-test equal variance | One tailed *T*-test unequal variance | Two tailed *T*-test (default) | Two tailed *T*-test equal variance | Two tailed *T*-test unequal variance |
|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | P-Value | P-Value | P-Value |
| **BLUE_AVG** | Precision | 0,01306955 | 0,032813886 | 0,020300573 | 0,0261391 | 0,065627772 | 0,040601146 |
| **ROUGE_L** | Recall | 0,115993637 | 0,005404706 | 0,158167038 | 0,231987274 | 0,010809412 | 0,316334075 |
| **METEOR** | | 0,064062754 | 0,008839869 | 0,115837589 | 0,128125508 | 0,017679738 | 0,231675179 |
| **CIDER** | | 0,454664135 | 0,156151563 | 0,287884006 | 0,909328271 | 0,312303126 | 0,575768012 |
| **SPICE** | F1 | 0,326144181 | 0,081495923 | 0,244329314 | 0,652288363 | 0,162991847 | 0,488658628 |
| **BERTscore** | | 0,276241506 | 0,025180214 | 0,18807362 | 0,552483012 | 0,050360428 | 0,37614724 |

Table 7.41: P-Values and Cohen's d for different types of statistical tests between 1-NN (simple) and 1-NN with Retrieval Augmented Generation submissions

| | *Metrics* | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **BLUE_AVG** | Precision | 0,928965346 | 0,349267679 | 0,698535358 | -1,29027074 | 0,219289013 | 0,438578026 | 0,843819825 |
| **ROUGE_L** | Recall | 0,943578011 | 0,349267679 | 0,698535358 | 0,387058511 | 0,219289013 | 0,438578026 | 0,843819825 |
| **METEOR** | | 0,992822543 | 0,349267679 | 0,698535358 | -0,024973978 | 0,219289013 | 0,438578026 | 0,843819825 |
| **CIDER** | | 0,873424259 | 0,349267679 | 0,698535358 | 0,479625319 | 0,219289013 | 0,438578026 | 0,843819825 |
| **SPICE** | F1 | 0,795779303 | 0,349267679 | 0,698535358 | 0,64597824 | 0,219289013 | 0,438578026 | 0,843819825 |
| **BERTscore** | | 0,868173347 | 0,349267679 | 0,698535358 | 0,885937588 | 0,219289013 | 0,438578026 | 0,843819825 |

Table 7.42: P-Values and Cohen's d for different types of statistical tests between 1-NN (simple) and Pegasus abstractive summarization algorithm submissions

| | *Metrics* | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **BLUE_AVG** | Precision | 0,848491288 | 0,033376508 | 0,066753015 | -2,597920072 | 0,022750132 | 0,045500264 | 0,032621652 |
| **ROUGE_L** | Recall | 0,002325612 | 0,066807201 | 0,133614403 | -1,886582377 | 0,047790352 | 0,095580705 | 0,114713137 |
| **METEOR** | | 0,227339605 | 0,033376508 | 0,066753015 | -2,67277122 | 0,022750132 | 0,045500264 | 0,032621652 |
| **CIDER** | | 0,115525632 | 0,433816167 | 0,867632335 | -0,022107134 | 0,5 | 1,00E+00 | 0,682310468 |
| **SPICE** | F1 | 0,04009871 | 0,433816167 | 0,867632335 | 0,0883232 | 0,5 | 1,00E+00 | 0,682310468 |
| **BERTscore** | | 0,032382713 | 0,433816167 | 0,867632335 | -0,058825661 | 0,5 | 1,00E+00 | 0,682310468 |

Table 7.43: P-Values and Cohen's d for different types of statistical tests between 1-NN with Retrieval Augmented Generation and 1-NN with Pegasus abstractive summarization submissions

| | *Metrics* | *F*-test | One tailed Mann-Whitney *U*-test | Two tailed Mann-Whitney *U*-test | Cohen's *d* - Baseline Effect Size test | One tailed Wilcoxon signed-rank test | Two tailed Wilcoxon signed-rank test | Kolmogorov-Smirnov test |
|---|---|---|---|---|---|---|---|---|
| | | P-Value | P-Value | P-Value | Cohen's *d* | P-Value | P-Value | P-Value |
| **BLUE_AVG** | Precision | 0,768768511 | 0,033376508 | 0,066753015 | -1,835426139 | 0,022750132 | 0,045500264 | 0,032621652 |
| **ROUGE_L** | Recall | 0,00347515 | 0,033376508 | 0,066753015 | -2,973656871 | 0,022750132 | 0,045500264 | 0,032621652 |
| **METEOR** | | 0,23368871 | 0,033376508 | 0,066753015 | -2,645711939 | 0,022750132 | 0,045500264 | 0,032621652 |
| **CIDER** | | 0,201912386 | 0,308537539 | 0,617075077 | -0,900570675 | 0,252492538 | 0,504985075 | 0,682310468 |
| **SPICE** | F1 | 0,119328716 | 0,308537539 | 0,617075077 | -1,298031071 | 0,252492538 | 0,504985075 | 0,682310468 |
| **BERTscore** | | 0,068301122 | 0,066807201 | 0,133614403 | -1,993577875 | 0,047790352 | 0,095580705 | 0,114713137 |