ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

School of Information Sciences and Technology
Department of Informatics
Athens, Greece

Master Thesis
in
Computer Science

# Exploring Diagnostic Captioning Methods

Vasilis Karatzas

*Supervisors:* John Pavlopoulos

Ion Androutsopoulos

October 2021

**Vasilis Karatzas**

*Exploring Diagnostic Captioning Methods*

October 2021

Supervisors: John Pavlopoulos, Ion Androutsopoulos

**Athens University of Economics and Business**

School of Information Sciences and Technology

Department of Informatics

Natural Language Processing group, Information Processing Laboratory

Athens, Greece

# Abstract

Image captioning has been researched a lot recently, but not much of that research has been applied to the biomedical domain. Diagnostic Captioning, the process of predicting diagnoses for medical images, can be very helpful for medical experts, since writing a diagnosis can be time-consuming and there is a lot of demand for it. In this master thesis the behavior of three types of models for diagnostic captioning is studied: image unaware unaware, retrieval, and image encoders combined with language models. The thesis also contains important findings on the difference that the preprocessing of the test captions can make in evaluation scores. Finally, this thesis concerns the participation of AUEB's NLP Group in the 2021 ImageCLEFmedical Caption competition, where the main driver was the author. The team earned the 2nd place among 8 teams with a retrieval based model.

# Περίληψη

Το πεδίο της παραγωγής περιγραφών εικόνων (Image Captioning) έχει ερευνηθεί αρκετά τελευταία, αλλά δεν έχει εφαρμοστεί πολλή από αυτήν την έρευνα πάνω στον βϊοιατρικό τομέα. Η παραγωγή διαγνωστικών περιγραφών εικόνων (Diagnostic Captioning), η διαδικασία πρόβλεψης διαγνώσεων για ιατρικές εικόνες, μπορεί να βοηθήσει αρκετά τους γιατρούς που κάνουν διαγνώσεις, καθώς η συγγραφή διαγνώσεων απαιτεί μερικές φορές αρκετή ώρα, και υπάρχει μεγάλη ανάγκη για υποστήριξη των γιατρών. Σε αυτήν την μεταπτυχιακή εργασία παρατηρούμε τη συμπεριφορά τριών τύπων μοντέλων για παραγωγή διαγνωστικών περιγραφών εικόνων: μοντέλα χωρίς γνώση της εικόνας, μοντέλα ανάκτησης, και κωδικοποιητές εικόνας σε συνδυασμό με γλωσσικά μοντέλα. Κάνουμε επίσης σημαντικές παρατηρήσεις σχετικά με τη διαφορά που μπορεί να κάνει η προεπεξεργασία των κειμένων στις βαθμολογίες. Συμμετείχαμε επίσης στον διαγωνισμό ImageCLEFmedical Caption του 2021, όπου πήραμε τη 2η θέση μεταξύ 8 ομάδων με μοντέλο βασισμένο στην ανάκτηση.

# Acknowledgements

I would like to thank professor Ion Androutsopoulos, senior lecturer Ioannis Pavlopoulos and research assistant Vasiliki Kougia for guiding me throughout my thesis and being considerate about a personal problem I had to face. I am very grateful for this team, as they are kind people and passionate about their work. I am also thankful for our cooperation with Foivos Charalampakos in a small part of this thesis.

I would also like to thank my mother (Neli Hristova), my cousins (Daniela Hristova and Ivan Hristov), and my best friend (Lia Kontakou), for helping me become the person I am today.

# Contents

# 1 Introduction

Diagnostic captioning (DC) is the process of predicting diagnoses, in the form of text, for medical images [Pav+21]. Writing diagnoses can be very time-consuming, since the medical expert needs to examine the image carefully, and sometimes has to combine the image with patient data, like the patient's history. Automatically generated diagnoses can speed up this process, since the medical expert will have a guide on what the diagnosis should be, and if the captioning model is very good at diagnosing, it might need little to no changes in its predictions.

This thesis concerns the participation of the author in the ImageCLEFmedical Captioning task of 2021 as member of AUEB's NLP Group.[1] [2] Furthermore, the research extends from the campaign, to the benchmarking of a variety of models in diagnostic captioning.

This thesis concerns three types of models:

1. **Image unaware language models**, which learn to predict an unseen caption by being trained on several seen ones (training set) while they completely disregard the images. At the time of inference, no information of the exam in question is being used. Hence, these models are only used as naive baselines, though the scores of baselines can sometimes be surprisingly high. The transformer [Vas+17] based models GPT-2 [Rad+19] and GPT [Bla+21] Neo were used. BERT [Dev+19] was also tried, but first experiments showed it was outperformed, although that should not mean that it is not as strong, but might need more tuning.

2. **Retrieval models**, which include the use of the $k$-nearest neighbors ($k$-NN) algorithm, an approach where, for each test image, the k closest training images, according to a similarity function, are retrieved, and their captions are used to form the prediction. This approach ended up working best with k=1, so the models will be referred as 1-NNs. The similarity function is given the outputs of an encoder for all the images.

3. **Encoder-decoder models**, which include state-of-the-art (SOTA) models in the captioning field, both biomedical and not; VisualGPT [Che+21], R2Gen [Che+20],

[1] The task is the *Caption Prediction Task*: `https://www.imageclef.org/2021/medical/caption` (Accessed: 19 October 2021)
[2] AUEB's NLP Group website: `http://nlp.cs.aueb.gr/` (Accessed: 19 October 2021)

and M2T [Cor+20]. VisualGPT and R2Gen are SOTA models in diagnostic captioning, while M2T is a SOTA model in generic image captioning, and was never tested before, to the best of the author's knowledge, on diagnostic captioning. All three models mentioned utilize the transformer [Vas+17] architecture which uses an encoder-decoder architecture approach.

## 1.1 Contributions

1. Evaluating and comparing (benchmarking) DC models in two different datasets. Some of these models have already been used for DC, but it's important to compare them to gain insight about what methods work better.

2. Applying a SOTA captioning model to DC, that to the best of the author's knowledge, was never used as a DC model before. Many captioning models are tested for the generic captioning task in the COCO [Lin+14] dataset and some of them use object locations which COCO provides. The object locations can be considered patches of the image that assist the model on its prediction by highlighting important areas. In this thesis, code was used to transform any image into the format of COCO; instead of object locations, the image is split into random patches. The random patches don't necessarily hold information about objects, so they won't actually assist the prediction. They are only used if the architecture required object locations.

3. Important points about preprocessing and scoring functions. How they can promote less comprehensible captions and how everyone should state their exact preprocessing steps when presenting scores, especially when comparing models. Big score differences can come from small alterations. Preprocessing differences are observed in two different times; before training and right before testing. The kind of preprocessing seen later involves lower-casing, the removal of punctuation and common words (stopwords), and stemming.

# 2 Background and Related Work

## 2.1 Required Background

To better comprehend this thesis, some knowledge of machine learning (ML) is required. This knowledge involves the process of training an ML model, and what neural networks are, along with some well-known kinds of neural networks; convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The reader may want to consult [Cho17] for a practical introduction to machine learning and deep learning in particular. Additional information for the following is given throughout the thesis: the encoder-decoder architecture, attention in ML, the transformer [Vas+17] architecture, the similarity functions used in this work, the $k$-nearest neighbor algorithm, beam search, and the score functions used for evaluation.

## 2.2 Some Additional Background

Before describing any models, some concepts will be given to better understand the model architectures. These concepts involve the encoder-decoder architecture, attention, and the transformer [Vas+17]. The encoder-decoder is an architecture with two main modules, one that produces the representation of the input (the encoder module), and one that takes that representation to produce the outputs of the model (the decoder module), as seen in Figure 2.1. The encoder-decoder architectures are separated in different classes, depending on how many times the encoder and the decoder modules are used. For example, when dealing with text, the input may be fed to the encoder word by word, meaning the encoder is used multiple times for one input. The classes of encoder-decoder architectures are: *Many to One*, *One to Many*, *and Many to Many*. This thesis only concerns the *One to Many* class, because the inputs are images, so they are given to the encoder as is, and the outputs are texts, so they are produced token by token. An example of the One to Many class can be seen in Figure 2.2.

Attention is a mechanism that suggests certain parts of data or representations as more useful. Through the use of attention weights, the most significant chunks of data, according to the attention mechanism, are influencing more the output of the model. The idea is that not all data are always important. More specifically, in image captioning, the model can

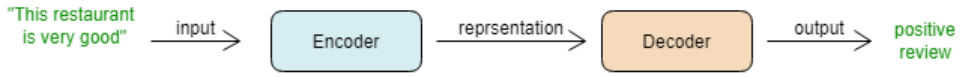


**Fig. 2.1:** An example of an encoder-decoder model used to classify positive or negative reviews of restaurants.

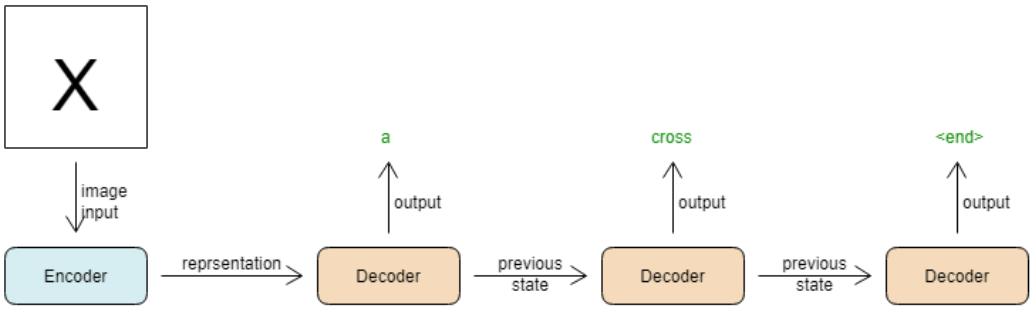


**Fig. 2.2:** An example of a *One to Many* encoder-decoder model used to predict a descriptive text for an image. The decoders share the same trainable weights.

focus on different parts of the image when predicting next words for the caption text, as shown in Figure 2.3.

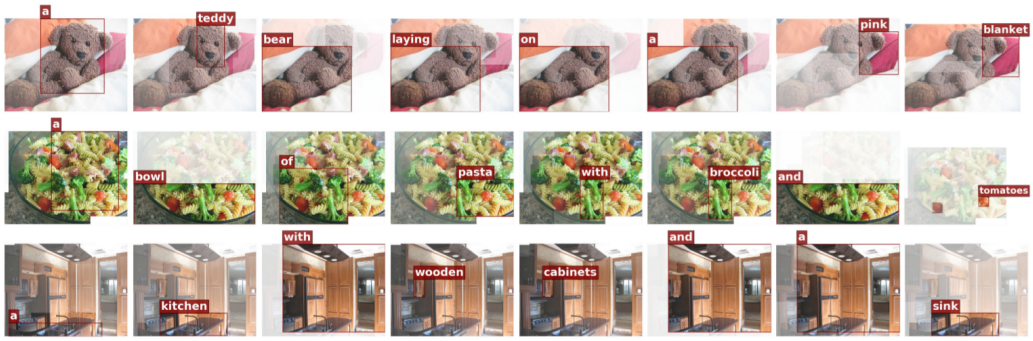


**Fig. 2.3:** An example of an attention mechanism in image captioning. At each new word to be predicted, the model focuses on a different part of the image, shown by the red rectangles, and partially ignores other parts, shown by the whiteness. This figure is from [Cor+20].

There is a variety of attention types, including self-attention, multi-head attention [Vas+17] and others [Cha+19]. These will not be described, as a basic knowledge of attention is enough to understand the models of this thesis.

The transformer is a popular encoder-decoder structure that utilizes self-attention. Its architecture can be seen in Figure 2.4. The left module of the picture is an encoder layer and the right module is a decoder layer. The encoder of the transformer has six of these encoder layers, and the decoder has six of these decoder layers. Only the output of the final encoder layer is passed to the decoder layers. The *Add & Norm* blocks of the figure

represent residual connections, by adding the matrices of a previous layer to the current one, followed by the corresponding normalization of that layer.

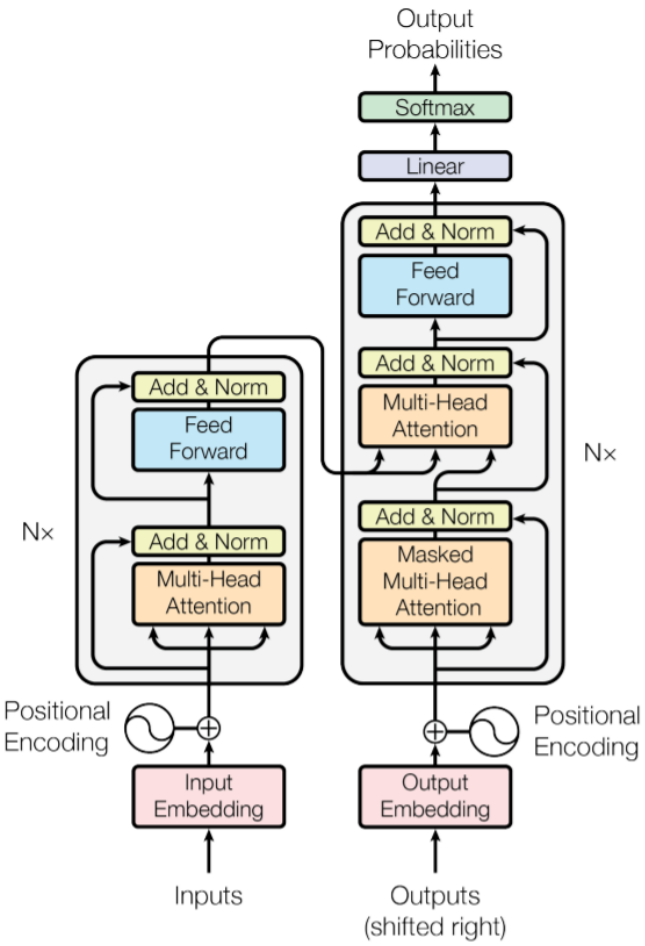


**Fig. 2.4:** The transformer architecture. This figure is from the original paper that introduced this architecture [Vas+17].

## 2.3 Generic Image Captioning

Generic image captioning is the task of predicting (generating) text that describes the content of images. One of the most well known datasets, if not the most well known, for evaluation of image captioning models is COCO [Lin+14]. COCO is a large dataset of 330k images, more than 200k of which have captions, and for each captioned image, five different captions are provided. COCO is not exclusively an image captioning dataset, since it also contains data about objects in the images, which can be used for object segmentation or object detection tasks. Many state-of-the-art (SOTA) models in the image captioning task of COCO also use the information of the object locations and classes, meaning they may not perform that well in another dataset that doesn't provide these.

Current SOTA approaches in COCO involve the M2 Transformer [Cor+20] and OSCAR [Li+20]. The M2 Transformer (M2T) is based on the transformer [Vas+17], so its architecture is very similar to that of Figure 2.4, with some variations. Instead of using the output of the last encoder layer, M2T combines the outputs from every encoder through an attention mechanism. The authors of the M2T paper [Cor+20] named this machanism *Meshed Cross-Attention*. The other variation that the M2T proposes is the addition of, what the authors call, *Memory-Augmented Attention* in its encoders. When attending to the input, the model adds trainable matrices to the attention mechanism. The main idea of these matrices is that by making them trainable they will be able to hold a priori knowledge from previous runs. More about the M2T architecture will be discussed in Section 3.3, since this model was benchmarked in this thesis.

OSCAR is not simply a model, but a mechanism for vision-language pretraining, where both text and images are given as inputs. The basis of this method can be observed in Figure 2.5. The input consists of a caption ($w$), object tags ($q$), and an image ($v$), while the training is done with two different losses; *Masked Token Loss*, where random tokens from $w$ and $q$ are masked and need to be predicted, and *Constrastive Loss*, where $q$ is randomly swapped (with 50% chance) with random other object tags from the dataset, and the model is asked to predict if $q$ was indeed swapped or not. This mechanism can be fine-tuned for many different task, as Figure 2.5 shows. The authors tested this mechanism by implementing an architecture that they also named OSCAR. The details regarding the architecture and its training are given in the caption of Figure 2.6, since some terms of that figure are used and it's better to read the details along with the figure. This thesis will not delve further to the model implementation, as the mechanism was the main idea.

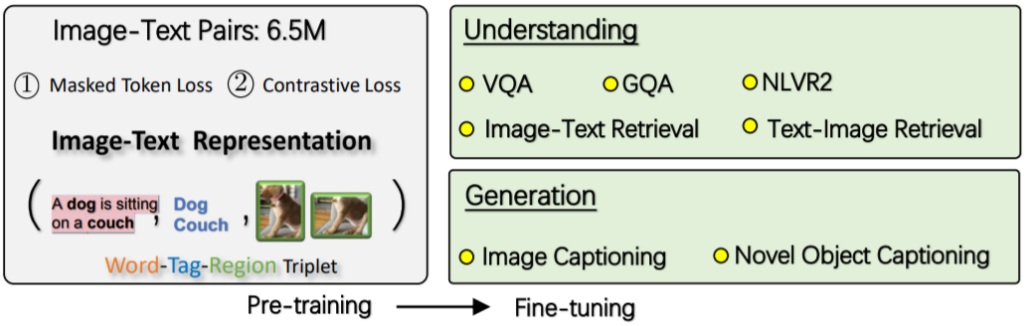


**Fig. 2.5:** The OSCAR method. Top left shows the two losses used for the two tasks mentioned in the main text. Bottom left shows the three different types of outputs. On the right, tasks that the model can be used on with fine-tuning can be seen. They are separated by the authors to *Understanding* and *Generation* tasks. This image is from [Li+20].

Finally, a model that used to be the SOTA approach in image captioning is Show, Attend and Tell [Xu+15]. The Show, Attend and Tell (SAT) model architecture is simple and can be seen in Figure 2.7. The input image is passed through a convolutional neural network (CNN) and the output of that network is used to generate tokens through a recurrent neural

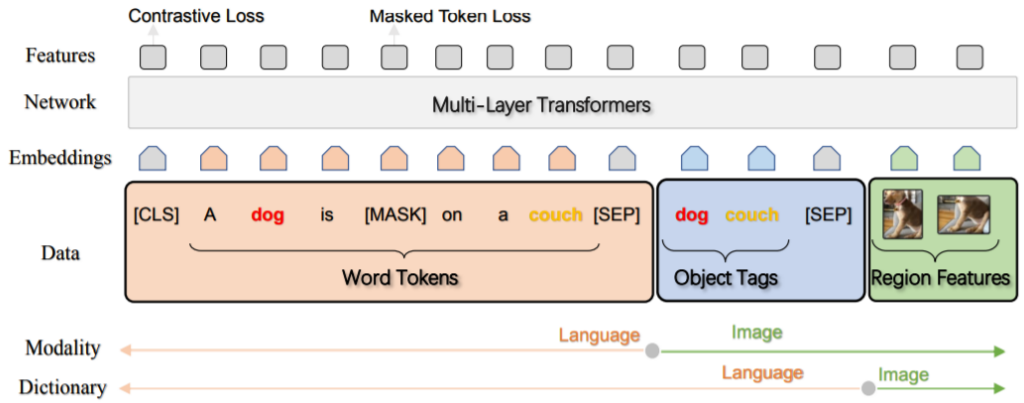


**Fig. 2.6:** The OSCAR model. The *Masked Token Loss* is calculated based on the *Dictionary* split. Its *Image* representation is used to predict the masked tokens of the *Language* representation. Similarly, the *Contrastive Loss* is calculated based on the *Modality* split. Its *Language* representation is used to predict if the *Image* representation was randomly swapped with another one from the training set. This image is from [Li+20].

networks (RNN), more specifically, a long short-term memory (LSTM) [HS97]. At each token prediction, the image is attended depending on the previous predicted tokens.

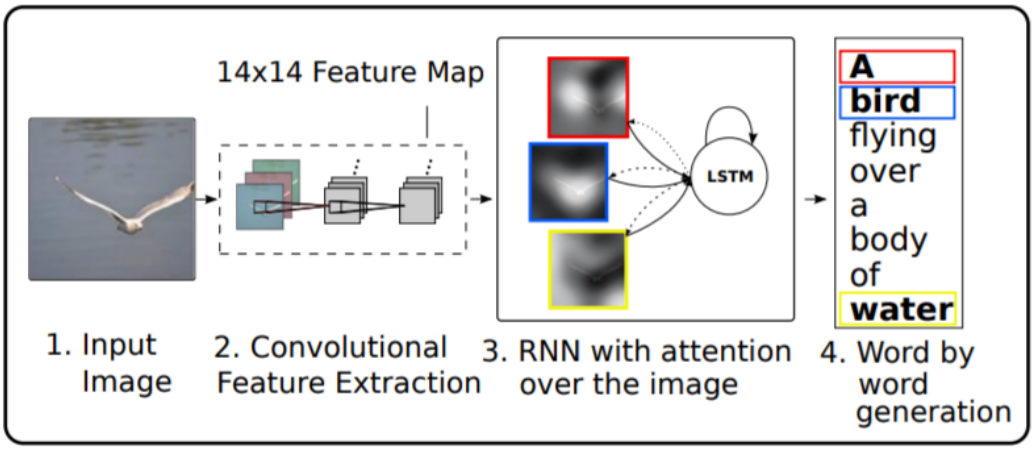


**Fig. 2.7:** The SAT model. This image is from [Xu+15].

## 2.4 Diagnostic Captioning

Generic image captioning has been studied a lot in recent years, but not many of those studies are applied to the biomedical domain. As mentioned in [Pav+21], diagnostic captioning involves the prediction of text diagnoses from patient images. This thesis is heavily inspired by [Pav+21], which is a great introduction to the diagnostic captioning task, having references to models, datasets, metrics, and other practices around the task. Another introduction to diagnostic captioning, with implemented architectures, is [KPA19a].

A SOTA model in this task is [Yua+19] and it can be seen in Figure 2.8. The encoder is first pre-trained for a classification task of image observations (the *Chest Radiographic Observations* output in the figure was used in this step). The idea behind this pre-training is to assist the encoder to learn concepts about the images, before the main training. Following this idea, another pre-training is done to classify popular concepts (the *Medical Concepts* output was used in this step). All the outputs of the encoder are representations of the images. This is mentioned because their names can be misleading and they can be thought to be the output of the previous tasks. For its main training, the encoder is fed all the images of each patient and outputs their combined representation (*Multi-view Visual Features*). This representation is fed to a decoder to gain a representation of the images' context, which is then attended with the *Medical Concepts* output and fed to a final decoder for token prediction. The encoder is a CNN and the decoders utilize LSTMs.

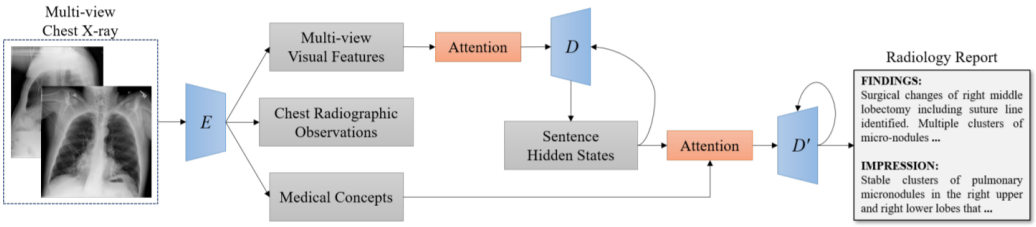


**Fig. 2.8:** A SOTA encoder-decoder model in DC. Module $E$ is an image encoder, module $D$ is a decoder for the representation of the images, and module $D'$ is the final decoder that uses the decoded representation of the images attended by the medical concepts. This image is from [Yua+19].

Until now, many of the aforementioned described models combined the captioning task with another one, usually a classification task. This is actually a common practice. Another example of this practice in DC involves RTEx [Kou+21], which can be seen in Figure 2.9. RTEx@R is a classifier for abnormalities; images are captioned only if they considered by the model to contain abnormalities. RTEx@T is a multi-label classifier, that chooses which tags are correct findings for the image. RTEx@X uses the images to create their final representation, which will be associated with the predicted tags. All RTEx modules use DenseNet [Hua+17], a famous CNN architecture with residual connections. The first two modules (RTEx@R and RTEx@T) add a classification layer on top, while the last one (RTEx@X) just outputs a representation. At inference time, the representation is compared with the representations of every training image that had the same tags, and the caption of the closest one is used as the final output. If no training image had the same tags, all training images are used. Later (in Section 3.2) the details of how the distance between image representations can be obtained will be discussed.

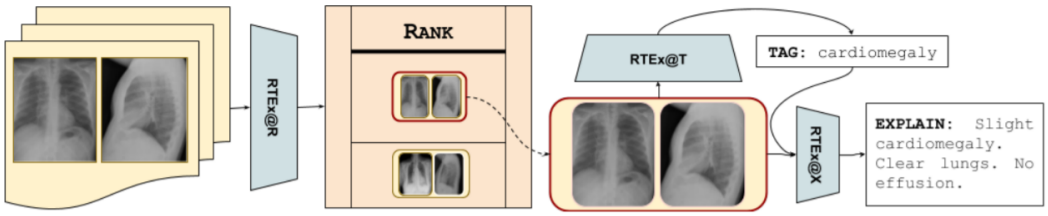


**Fig. 2.9:** RTEx, a DC model that combines DC with tagging and abnormality classification. This image is from [Kou+21].

## 2.5 Predictive Text

Another related task to this thesis is predictive text. In this thesis, this term means a mechanism that suggests the continuation to an incomplete text. Instead of captioning an image, a system can assist an author trying to describe an image by suggesting the next word to their text. This is a very close task, since the only differences are: A) having also text as input (the text already written by the user) apart from the images and B) predicting only the next word at a time, which can be easily implemented in most captioning models, since they already predict text. Although they assist the writer, there has been an interesting study [ACG20], which shows that predictive text can change one's typing behavior.

Clinical predictive text suggests the next word to an incomplete diagnosis instead. In [PP20], two approaches for clinical predictive text were benchmarked. One was based on $n$-grams, predicting the next word of an incomplete text based on the most common next word for that incomplete text in a corpus. Instead of using the whole previous text, only the $n$ previous words were used. The other approach was RNNs. Both long short-term memory (LSTM) [HS97] and gated recurrent unit (GRU) [Cho+14a] were used. These approaches only had the text as input though.

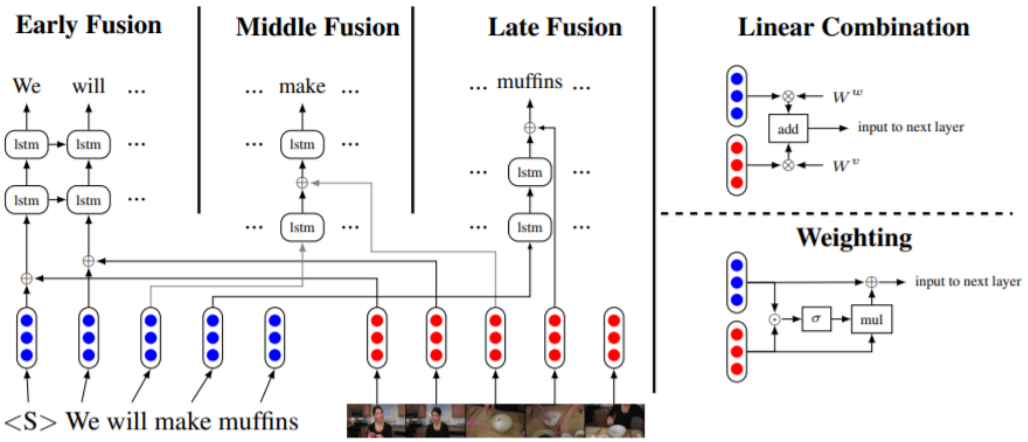


**Fig. 2.10:** Different ways of combining visual with text data. This image is from [AKL19].

Having inputs of more than one types of data (for example images and text written so far) makes an architecture multimodal [KSZ14]. In [AKL19], some methods of combining image with text data in language models can be seen (also seen in Figure 2.10). For a stacked RNN, the different types of data can be combined before their insertion to the RNN layers (*Early Fusion*), between RNN stacked layers (*Middle Fusion*), or after one type of data has passed through the RNN (*Late Fusion*). The data can be combined through an addition of their weighted representations (*Linear Combination*) where the weights are trainable, or a concatenation of the text representation with a weighted version of the visual representation (*Weighting*), where the weighting mechanism uses the text representation.

## 2.6 ImageCLEFmedical Campaign

Part of this thesis involves the participation of the author in the CLEF 2021 campaign [1] [2]. More specifically, the author competed as a member of AUEB's NLP Group in the ImageCLEFmedical [Ion+21] Caption task of 2021.[3] ImageCLEF is an evaluation campaign that has been held annually since 2003, and it involves a number of tasks that are related to images.

The models for the participation were heavily inspired from previous participations of AUEB's NLP Group in the same campaign of past years. The papers of these past participations are [KPA19b] and [Kar+20], while [KPA20] was also related to these models. From these papers, the model that was used for this year by the author is the $k$-nn based approach. This will be discussed in more detail in Section 3.2, but the main concept of it is that the image is encoded and then the captions of the $k$ closest images in the training set are retrieved so that they are combined for the final result. This year, more encoders were tested, and two different functions to calculate the distance of the representations were used.

## 2.7 Models Considered After ImageCLEFmedical

Beyond the campaign, VisualGPT [Che+21], R2Gen [Che+20], M2T [Cor+20], and an architecture similar to Show, Attend and Tell [Xu+15] were benchmarked in this thesis. VisualGPT and R2Gen are also based on the transformer architecture, like previously mentioned models. Regarding the alterations from the transformer architecture, VisualGPT replaces a residual connection in the decoder layers with gates and R2Gen replaces all

[1] CLEF: `http://www.clef-initiative.eu/` (Accessed: 19 October 2021)
[2] CLEF 2021: `http://clef2021.clef-initiative.eu/` (Accessed: 19 October 2021)
[3] More information about the task can be found at `https://www.imageclef.org/2021/medical`

residual connections in the decoder layers with a module that uses a memory matrix. Additional information for the gates and the memory module is provided in Section 3.3.

Some models used in this thesis are retrieval based models, while the rest of the approaches are language models. A good introduction to the methods used for retrieval is [Sin01], while the exact retrieval method for this thesis will be described in Section 3.2. In general, many language models for captioning follow the encoder-decoder ([Cho+14b], [BKC17]) architecture, and on that matter, many encoder-decoder models are inspired by, or use, the transformer [Vas+17] architecture, as it was also observed in previously mentioned models.

Some models use patches of the images for their inputs. This is mainly because models tested in COCO usually take advantage of its data about object locations. To make these models work, random patches were created, since object locations were unknown in other datasets. Altering the task's type of inputs is not an uncommon strategy in many machine learning (ML) tasks, and there is even an example of this for captioning, where the inputs are also questions with answers for the image [Fis+20]. Breaking the image to patches sort of resembles how humans can look at a certain part of an image before they write about it, when describing an image, and there is even a model that actually uses the gaze of the human as input [Tak+20].

Finally, in this thesis there are some ensembles of models. Ensembles are combinations of different models, and can be created in various ways, like summarizing the captioning outputs of several models, or using the caption that most models predicted. In [VKL20], where four popular language models are tested for text prediction, an ensemble approach was suggested. A new dataset is created where the input is the same as before (the previous text) and the output is a vector with four elements, one for each language model. If a language model predicted correctly the next word then the corresponding value is equal to 1, and otherwise it is equal to 0. Then a classifier is trained on that dataset. At inference time the input is given to the classifier that decides which models might predict correctly the next word, and from these models, the prediction is taken from the model that was more confident about its prediction (through the probability given to the chosen word).

# 3 Methods

This section concerns the models used for the participation of AUEB's NLP Group in the 2021 ImageCLEFmedical Caption task, as well as other diagnostic captioning models that were trained after the campaign.[1] [2]

## 3.1 Image Unaware Language Models

These are simplistic baselines, implemented for the captioning task, which do not take into account the images of the corresponding captions. Each of them was trained as a language model using all of the training captions as data. The models used were GPT-2 [Rad+19] and GPT Neo [Bla+21].[3] At train time, the models were fed with the training captions, and at test time, the models were used to generate text without utilizing any of the images.

In text generation, many models form their prediction token by token. The next token can be the most probable one to continue the previous text, but this way the model may predict similar texts, and the generated token sequence may not be the globally (end-to-end) most probable one. To impose some randomness and to search for the globally most probable sequence, instead of picking the most probable next token, beam search can be performed. In beam search, to predict a token, the $k$ most probable tokens are gathered, and the prediction continues, using beam search again, either until all $k$ branches reach the ending token or for some predefined steps. Some branches can be cut off, if the other branches have more probable outcomes, since only $k$ branches must be kept at each step, but $k \cdot k$ are produced. The $k$ variable is also called the beam size. For each token prediction of the language models shown in this thesis, beam search was used with a beam size of 3.

An example can be seen in Figure 3.1. In that example, it can be observed that "a" will be the chosen next token of the text "the cat is". If the choosing mechanism was not beam search, and instead the chosen next word was the most probable word after "the cat is", then "the" would be chosen instead, since it has a higher probability (16%) than "a" (15%). Beam search is used to get closer to the globally most probable token sequence, and not

[1] The task is the *Caption Prediction Task* here: `https://www.imageclef.org/2021/medical/caption` (Accessed: 19 Octoberr 2021)
[2] AUEB's NLP Group website: `http://nlp.cs.aueb.gr/` (Accessed: 19 October 2021)
[3] `https://huggingface.co/transformers/master/model_doc/gpt_neo.html`

just the next token. In predictive text, greed selection of next tokens can be used, but in text generation beam search is usually preferred for the aforementioned reason.

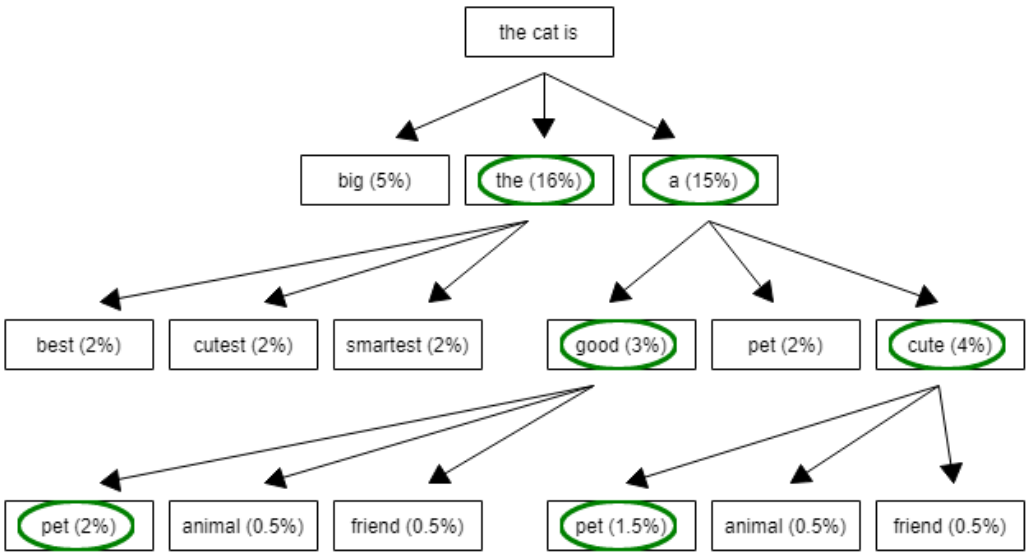


**Fig. 3.1:** An artificial example run of beam search with beam size = 2. From each word, the three most probable next words are shown. The chances in the parentheses are the overall chances of the sentences up to that point. Even though "the" is chosen at first (along "a"), at the next step, some branches from "a" have higher overall chance than every branch of "the", so they are chosen instead. At the end, one branch will be chosen, and the starting token of that branch will be the next predicted token. In this example, "a" will be the next token of "the cat is", since no matter how deep the branches go, every branch originates from it.

In Table 3.1, the training hyperparameters of the models can be observed. It should be noted that two strategies for feeding all the training captions were tested. The first strategy is to create a separate input from each training caption by adding a starting token at the beginning of the caption and either cut tokens or add padding tokens to force the same length of tokens to each input (see Strategy I in Table 3.2). For the second strategy, each training caption is tokenized and gets a starting token, then they are all merged and separated into inputs based on the input length. In Strategy II of Table 3.2, it can be observed that the first input has parts from both training captions and that the only input with padding tokens is the last one. The second strategy performed better in early stages of development so this was kept. The better performance can be due to the fact that the first strategy enforces a big amount of padding tokens to the captions.

| Model | Huggingface name [4] | Epochs | Batch Size | Block Size | Optimizer | Learning Rate |
|---|---|---|---|---|---|---|
| GPT-2 | gpt2 | 15 | 12 | 52 | Adam | 3e-5 |
| GPT Neo | EleutherAI/gpt-neo-125M | 10 | 12 | 52 | Adam | 3e-5 |

**Tab. 3.1:** The hyperparameters of the baseline models. Take note that the batch size of these models was predetermined by their architecture and cannot be altered.

[4] `https://huggingface.co`

| Strategy | **Training Caption 1:** 'Left Upper lobe mass'<br>**Training Caption 2:** 'Duplicated Right Renal System' |
|---|---|
| **I** | Input 1: [start], Left, Upper, lobe, mass, [pad], [pad], [pad]<br>Input 2: [start], Duplicated, Right, Renal, phrase, [pad], [pad], [pad] |
| **II** | Input 1: [start], Left, Upper, lobe, mass, [start], Duplicated, Right<br>Input 2: Renal, System, [pad], [pad], [pad], [pad], [pad], [pad] |

**Tab. 3.2:** An example of the two strategies (I & II) used to create inputs, in an artificial training set with two captions and an input length of eight tokens. For simplicity, no preprocessing is performed.

## 3.2 Retrieval Methods

| Model | Input Shape |
|---|---|
| EfficientNetB0 | 224x224 |
| EfficientNetB7 | 600x600 |
| DenseNet121<br>DenseNet201 | 224x224 |
| InceptionV3 | 299x299 |
| ResNet50<br>ResNet152V2 | 224x224 |
| NASNetLarge | 331x331 |
| InceptionResNetV2 | 299x299 |
| Xception | 299x299 |

**Tab. 3.3:** Image encoders and image input shapes in the image-aware text generation models.

The following models were inspired by the baseline 1-NN model used in [Pav+21] and previous submissions [KPA19a; KPA19b; Kar+20] of AUEB's NLP Group in the ImageCLEFmedical Concept task (Section 2.6), which utilized the $k$-nearest neighbors algorithm ($k$-NN). The $k$-Nearest Neighbors algorithm is a retrieval based method that uses representations of data to calculate their distances/similarities with each other. At inference time, the representation of the input instance is created, and the $k$ training instances (examples) with the closest/most similar representations are selected as our sources to create the output. In the simplest versions of $k$-NN, which are intended to handle single-label multi-class classification, the input is assigned the majority label of the $k$ neighbours. But in the case of multi-label multi-class classification, where multiple labels can be assigned to an instance, more elaborate strategies are needed to obtain the labels of the input from the $k$ neighbours. Later, the strategies used in this thesis to combine the data from the $k$ neighbors are explained. Finally, in $k$-NN the data can be of any type; image, text, sound, video etc. An example of the algorithm's execution can be seen in Figure 3.2.

For these models, image encoders pre-trained on ImageNet [Den+09] were used to output an embedding (vector representation) for each image. Then, given a test image, the $k$

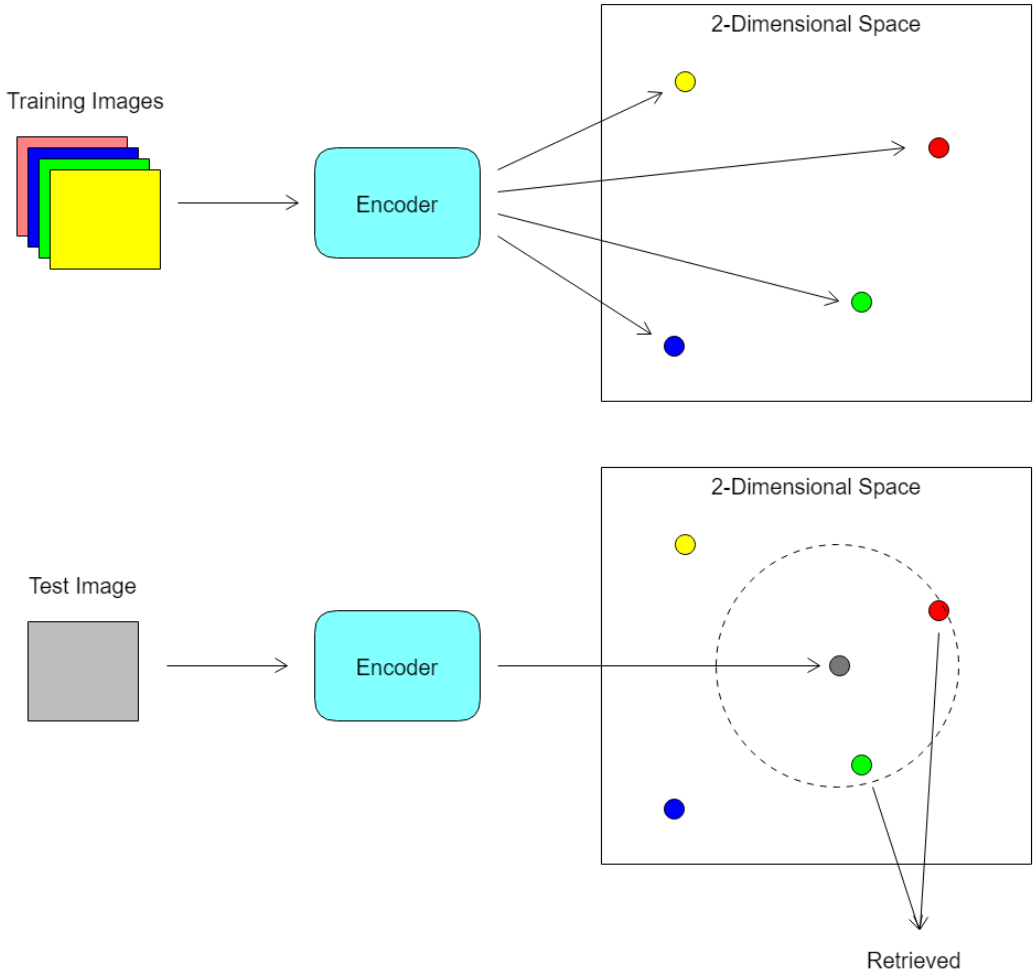


**Fig. 3.2:** An example run of the $k$-NN algorithm. It is assumed that $k = 2$, the representations of the images given by the encoder are two dimensional vectors, and that the Euclidean distance between points is used as the similarity function.

closest training images were retrieved, by using a function that calculated the similarity (or distance) between that image's embedding and every training image's embedding. For the function, cosine similarity was used, which performed better than matrix multiplication. Regarding two vectors, $u$ and $v$, the cosine similarity is calculated as follows:

$$\frac{u \cdot v}{||u||_2 \cdot ||v||_2}$$

where $u \cdot v$ is the dot product of $u$ and $v$. Regarding the same vectors ($u$ and $v$), the similarity by matrix multiplication is calculates as follows:

$$u_{norm} \cdot v_{norm}$$

where where $u_{norm} \cdot v_{norm}$ is the dot product of $u_{norm}$ and $v_{norm}$, and:

$$u_{norm} = \frac{u}{\sum_1^n u_i}$$

$$v_{norm} = \frac{v}{\sum_1^n v_i}$$

Note that before the images can be fed into an encoder, they have to be reshaped based of what the corresponding encoder excepts as input (see Table 3.3 for the needed shapes of the images). Regarding the combination of $k$ captions, from the retrieved images, the following approaches were used:

- Summarization. Converting a longer text into a smaller one. The longer text in this instance would be the concatenation of the retrieved captions. An existing summarizer was employed.[5] The idea of this summarizer is very simple, it keeps the sentences that contain the most relevant words of the text. The most relevant words are considered to be the most frequent ones in the whole text that are not stopwords. Summarizers like this, that don't generate new text but keep parts of the old one and combine those parts, are called extractive [All+17].

- Splitting every caption into sentences. Creating the output by using the $r$ most frequent sentences, where $r$ is a hyperparameter. This is meaningful only when the dataset has many repeated sentences across different captions, which the 2021 ImageCLEFmedical Caption training set had

These approaches did not yield better scores than simply selecting only the caption of the closest image as the output, meaning that in the following experiments $k$ was always equal to 1, thus the approach will be referred as as 1-NN.

Even though 1-NNs are simple models, they were outperforming many others in the ImageCLEF campaign, for both the captioning and concept tasks, and there was room for improvement. Ideas for improvement were as follows:

[5] `https://www.geeksforgeeks.org/python-text-summarizer/` (Accessed: 19 October 2021

- Tag-trained encoders. Since another member of AUEB's NLP Group was competing in the Concept task of the same campaign, some 1-NN models used that member's pre-trained encoders, which were trained to predict the appropriate medical tags of each image. For more information see Section 3.1.1 of [Cha+21].

- Ensembles of different 1-NNs. 3 or 5 1-NNs with different encoders were chosen and one caption from each of them was obtained. The final caption was the most frequent one amongst them. In case of ties, the caption of the best model (according to development set scores), amongst those that made the tie, was selected. Ways to combine the captions, like the ones previously mentioned, were also tried, but with no avail. It should also be mentioned that increasing $k$ from 1 for each model was also not useful. Finally, if every model gave a different output (a state called non-Agreement for the work of this thesis), there was an attempt to use GPT-2 to generate the caption instead, since GPT-2 was the second best approach, after 1-NNs, for the campaign, although GPT-2 generates the same sentence every time (not one that is an exact copy of a caption in the dataset).

## 3.3 Encoder-Decoder Models

While participating in the ImageCLEFmedical Caption task of 2021, two encoder-decoder models that utilized the image were used. The first one can be seen in Figure 3.3, and is a model similar to Show, Attend and Tell [Xu+15] regarding their use of attention and their decoder architecture.[6]

The main difference between the aforementioned model of the thesis and the Show, Attend and Tell model is that the latter uses its own convolutional neural network (CNN) for its encoding step, that splits the image in patches. The representations for the patches are gathered from a lower convolutional layer of the CNN. Another difference is that, although both architectures use recurrent neural networks (RNNs), the architecture of the participation uses a gated recurrent unit (GRU) [Cho+14a], while the Show, Attend and Tell architecture uses a long short-term memory (LSTM) [HS97].

The second encoder-decoder model tried is a token classifier that uses both the previous token predictions and the image to generate the next token (its architecture can be seen in Figure 3.4).[7] For this model, a training caption of $n$ tokens will result in $n$ different training samples, since the model will have a different training sample for each token prediction.

[6] `https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/text/image_captioning.ipynb`

[7] Inspired by: `https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8` (Accessed: 19 October 2021)

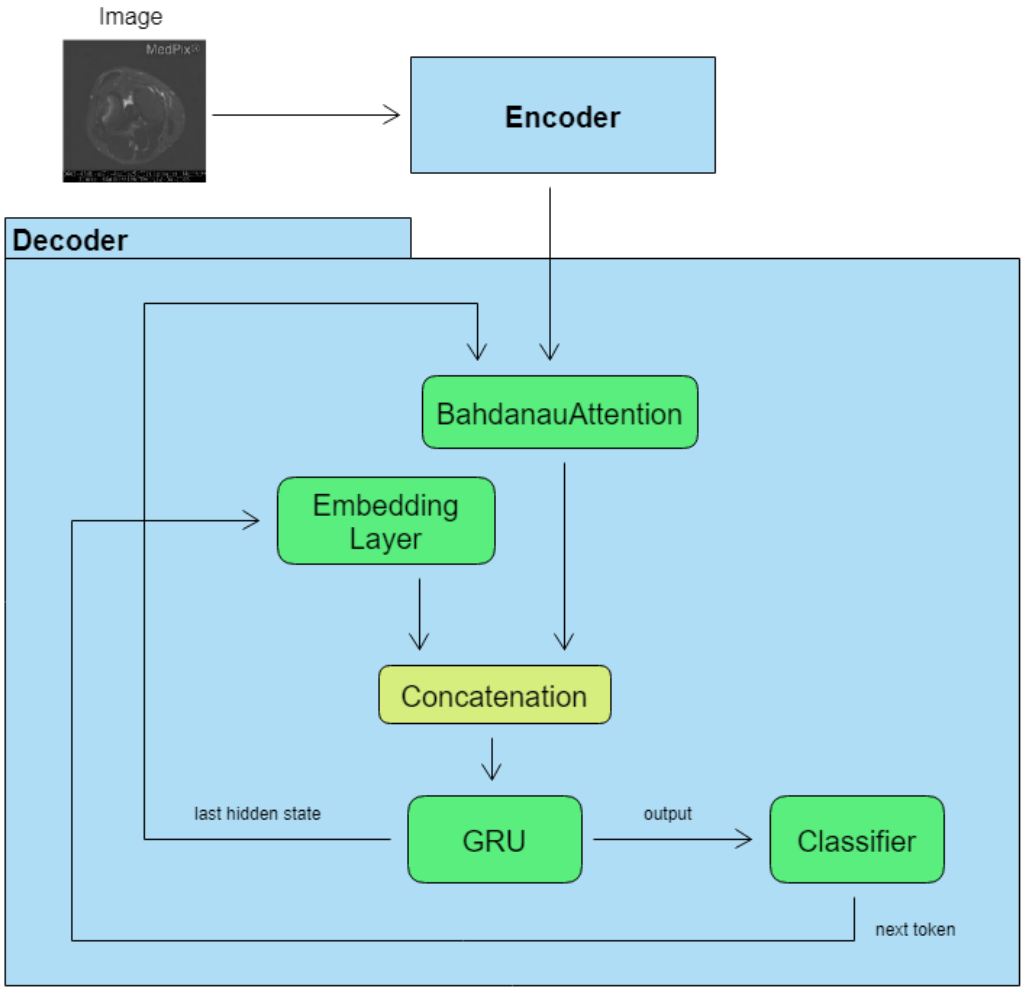


**Fig. 3.3:** The architecture of the Show, Attend and Tell inspired model. The image encoder used was InceptionV3 [Sze+16] and the image attention mechanism was based on [BCB15].

For each of these $n$ predictions, the input will also consist of the image, which is given to an encoder with 50% dropout and then a dense layer to generate the image representation as an embedding. The tokens are also represented as embeddings through a trainable embedding layer with 50% dropout. To generate the final representation of the previous tokens, the token embeddings are fed to a Long Short-Term Memory (LSTM). After that, the two final representations (the image and the tokens) are concatenated and given to a feed forward neural network (FFNN) with one hidden layer. This model resembles a predictive text mechanism, and it had an 8% accuracy when evaluated as a predictive text model, which was considered low by the author, so the development of this model stopped. In comparison, later on it is observed that a 6% accuracy predictive text model scored 10.9 in BLEU-4 (the ImageCLEF's metric that will described later), while the best model of this thesis for ImageCLEF had a BLEU-4 score of 55.342.

After the campaign ended, further experiments were done with the following state-of-the-art (SOTA) captioning models:

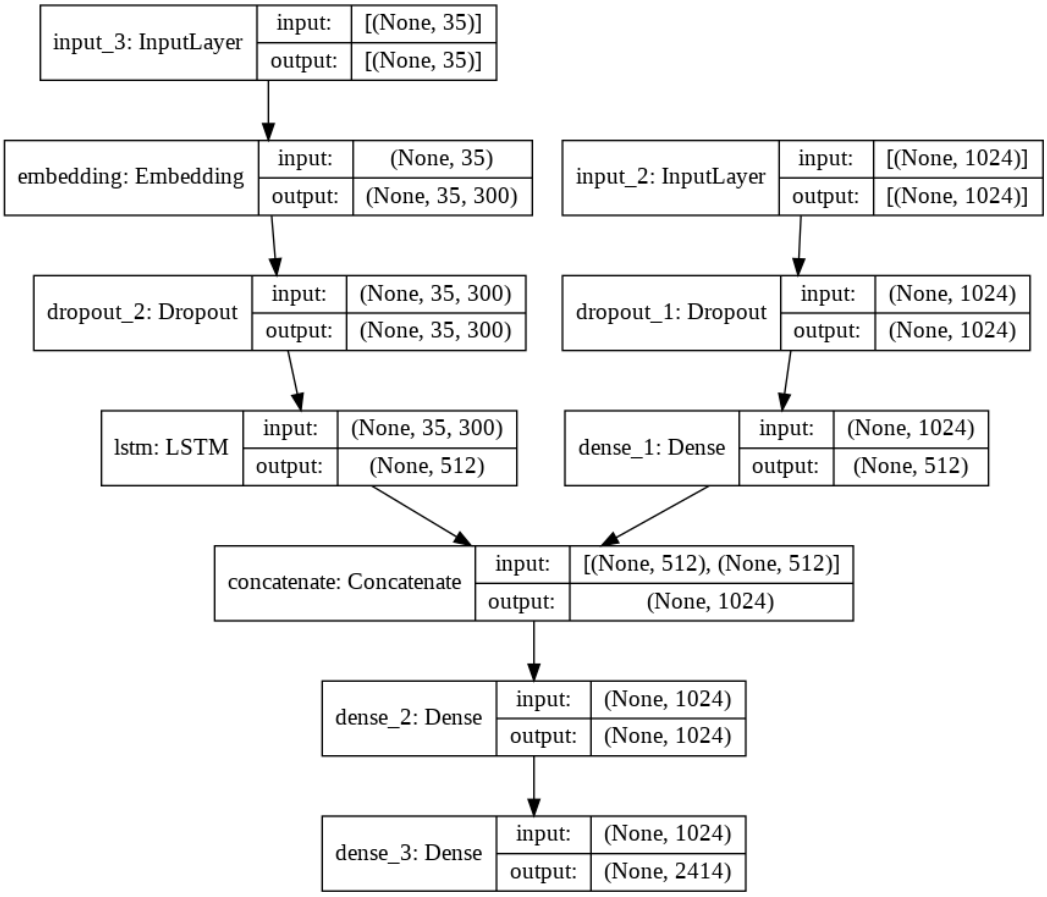


**Fig. 3.4:** The architecture of the second encoder-decoder model used. This was dropped since early experiments showed low scores.

- **VisualGPT** [Che+21].[8] SOTA when evaluated on the diagnostic captioning dataset IU X-ray.

- **R2Gen** [Che+20].[9] Previous SOTA when evaluated on the diagnostic captioning dataset IU X-ray (still holds the best score in one metric as will be observed later on).

- **M2T** [Cor+20].[10] SOTA when evaluated on the image captioning task of the COCO [Lin+14] dataset.

These models follow the encoder/decoder architecture where the image is given to the encoder unit in order to be represented as features, and then those features are passed to the decoder to output the predictions. They are also stacked, meaning that there are many layers of encoders/decoders instead of one. They share a similar decoder architecture, since their decoders are based on transformers [Vas+17].

[8] https://github.com/Vision-CAIR/VisualGPT
[9] https://github.com/cuhksz-nlp/R2Gen
[10] https://github.com/aimagelab/meshed-memory-transformer

A more detail view of the decoder architectures for the aforementioned models can be seen in Figure 3.5, which is a figure partially taken from [Che+21]. These decoder architectures follow the same three steps:

1. Step 1. Attending on the previous decoder output. Self attention is used in this step, which only uses one input (the previous decoder output) to decide where to attend on that input.

2. Step 2. Combining the result of the previous step with the image features. Cross attention is used in this step, which uses two inputs to decide where to attend in one of them. The result of the previous step is used to decide where to attend on the image features.

3. Step 3. Passing the output of the previous step through a feed forward neural network.

The *Add & Norm*, as also mentioned in Section 2.2, blocks are simply residual blocks (they add the matrices of a previous layer to the current one), followed by the corresponding layer normalization.

The following differences between the transformer decoder and the other decoders can be noted:

1. R2Gen adds a memory block that the authors named *Relational Memory*. The relational memory comprises a memory matrix that is updated each time a new token is predicted. The main idea behind this memory matrix is to hold information from the execution of previous token predictions. It is updated by a multilayer perceptron (MLP) with multi-head attention applied to the current memory matrix and the previous model outputs.

2. M2T adds an attention mechanism that uses the outputs of every encoder instead of the last one. The authors named this *Meshed Cross-Attention*. Each encoder output is attended through cross attention, using the previous decoder output as guide (that has already been attended through self attention). Then, a weight matrix named $\alpha$ that is calculated through trainable weights, is used to decide which attended encoder outputs are used.

3. VisualGPT, instead of using a residual connection after attending on the image features, it uses two matrices, $B^V$ and $B^L$, also known as gates, to control what information from the attended image and the attended previous decoder output will be used. Gate $B^V$ is multiplied element-wise with the attended image and $B^L$ is multiplied element-wise with the attended previous decoder output. In each

value of the gate matrices, exactly one gate is equal to zero, meaning that only one of the attended image or the attended previous decoder output is passed through. These gates hold weights that are not trainable, and are calculated from the attended previous decoder output. If the value at place $i$ of that output is below a certain threshold then $B_i^V = 0$ , while if that value is above that threshold then $B_i^L = 0$ . The main idea is that high values should be used to attend on the image, while low ones should be left as they were.

VisualGPT's and R2Gen's encoders are simple, using only an attention layer with a feed forward neural network (FFNN). M2T encoders follow the same architecture but they also have, what the authors call, *Memory-Augmented Attention*. Instead of simply using self attention, *Memory-Augmented Attention* resembles cross attention, meaning that it uses one source of data to decide where to attend on the other, but in *Memory-Augmented Attention* the sources are actually the same, and one is just augmented by memory matrices (the matrices get concatenated to the source data). The idea is, that by making the memory matrices trainable, they should eventually hold a priori knowledge (from previous executions). Another common attribute of these models is that they do not pass the image itself into the encoders, but rather a representation of the image that is split into patches. VisualGPT and M2T were mainly tested on the COCO dataset [Lin+14], which contains information about object places in images. Those objects could be given to the encoders as the patches of the image mentioned earlier. Since these models are not being tested on COCO for this thesis, a visual extractor of R2Gen was used, in order to create the patches of the other models.

Although VisualGPT was tested on a medical dataset (IU X-Ray), the available code for it was given for the COCO dataset, which was not working for IU X-Ray, and required new data loading classes. M2T has never been tested on IU X-Ray (or any medical captioning task, to the best of the author's knowledge). The hyperparameters for these models can be seen in Table 3.4.

| Model | Patience | Best Epoch | Batch Size | Optimizer | Learning Rate |
|---|---|---|---|---|---|
| VisualGPT | 5 | 9 | 25 | AdamW | 1e-14 |
| R2Gen | 50 | 15 | 16 | Adam | 5e-5 for visual extractor<br>1e-4 for rest |
| M2T | 5 | 11 | 50 | Adam | 1 with warmup 10,000 |

**Tab. 3.4:** The hyperparameters of the captioning models used for IU X-Ray. For Adam and AdamW see [KB14] and [LH18] respectively. For architecture-specific parameters please read the corresponding papers: [Che+21], [Che+20], and [Cor+20].

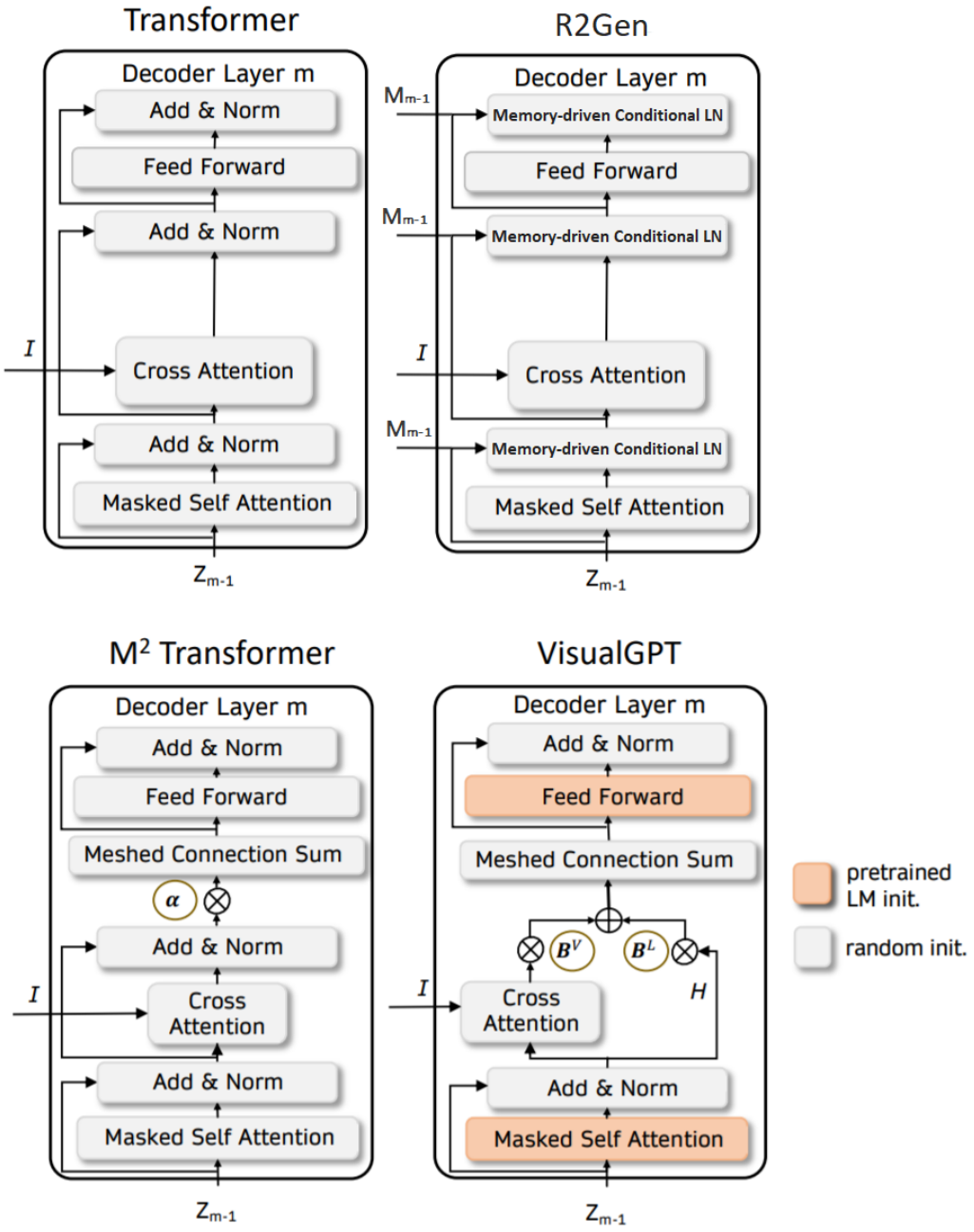


**Fig. 3.5:** A comparison of the decoders of the three transformer-based models. The transformer decoder is also shown on the top right. $I$ are the encoded image features, H are language features, $Z_{m-1}$ is the output of the previous decoder, and $M_{m-1}$ is the memory from the previous decoding step. Please note that this figure was taken (and partially augmented) from [Che+21].

# 4 Data

## 4.1 2021 ImageCLEFmedical Captioning

The 2021 ImageCLEFmedical Captioning dataset includes 3,700 radiology images, and their annotations made by medical experts. The images are identical to those in the 2020 ImageCLEFmedical VQA task [Aba+20], and they originate from the Med-Pix database.[1] Unlike the previous year, the modalities (e.g. X-Ray, MRI) of the images were not given for each image. For 444 (of the 3,700) images, the captions were not given to the participating teams, as these images would be used as the test set for the final results. The rest of the images were split into training and validation sets by the organizers, but we merged these sets to make new splits. The merged data was split into a training set (60% of the merged data), where the models were trained, a development set (20%), where the hyperparameters of the models were tuned, and a validation set (20%), where the models were tested based on the campaign's score function to decide the final submissions of the group.

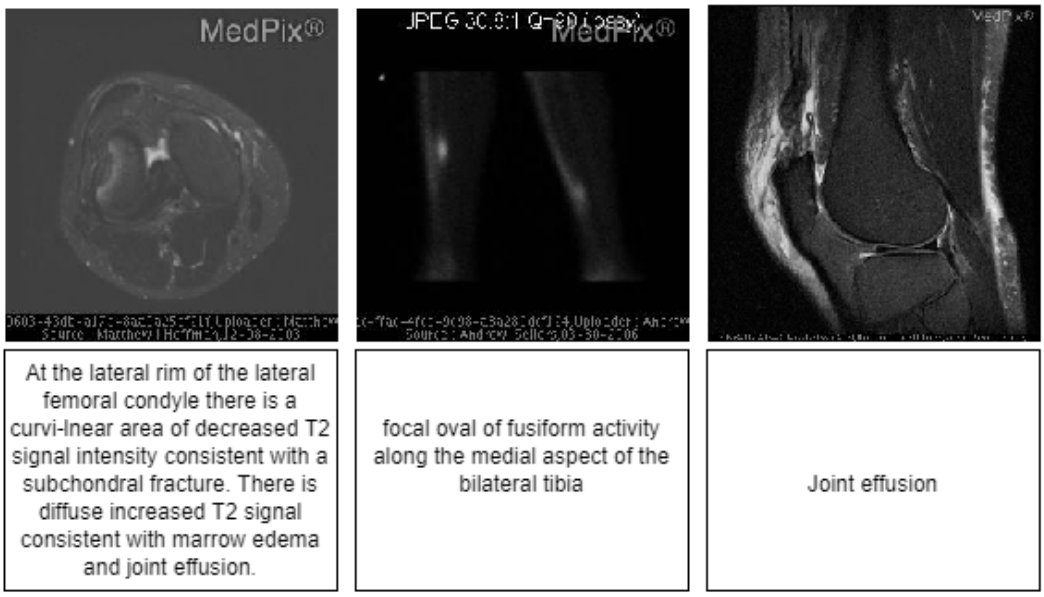


**Fig. 4.1:** Three random images from the 2021 ImageCLEFmedical Captioning dataset, along with their corresponding captions.

The maximum number of words in a caption was 43 (in 10 images), while the minimum number of words was 1 (also in 10 images). More about the distribution of caption lengths

[1] https://medpix.nlm.nih.gov/

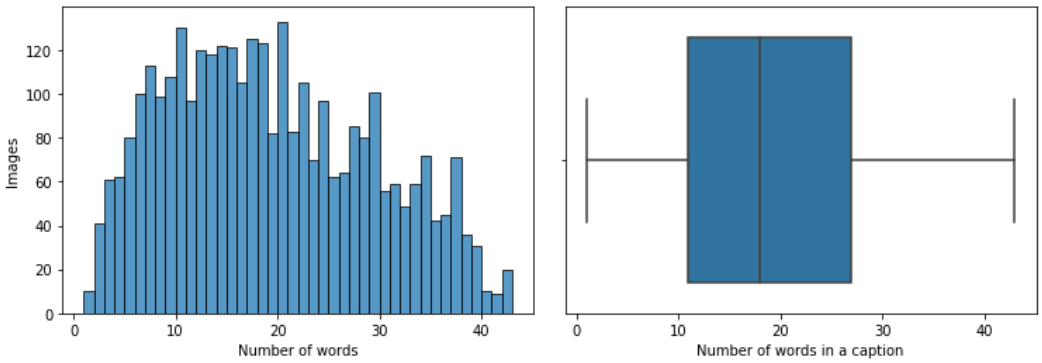


**Fig. 4.2:** Plots about the number of words in each caption. On the left, a histogram for the number of images with captions of a specific word length. On the right, a boxplot for the number of words inn captions.

can be seen in Figure 4.2, which doesn't seem to have any outliers. After lower-casing each word, the total number of distinct ones was 3,515, although 1,071 of them only appeared once. Table 4.1 shows the most common words in all the dataset (excluding the test subject). It was noticed that 1,141 captions (about 35% of captions) were duplicates (they were the caption of more than one image). Table 4.2 shows the most common captions. Having that many duplicates was an indication that retrieval models would most likely perform well.

| **Most common words w/ stopwords** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Word** | the | of | with | and | a | right | in | left | to | mass |
| **Occurrences** | 2,139 | 1,770 | 1,179 | 1,149 | 891 | 800 | 763 | 666 | 630 | 621 |

| **Most common words w/o stopwords** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Word** | right | left | mass | ct | demonstrates | axial | images | image | contrast | within |
| **Occurrences** | 800 | 666 | 621 | 616 | 511 | 451 | 385 | 379 | 365 | 302 |

**Tab. 4.1:** The 10 most common words of all captions, w/ and w/o stopwords.

The organizers stated that the final score would be calculated using the BLEU-4 metric, a variant of BLEU [Pap+02] (more about the metric will be discussed later. in Chapter 5). Before the calculation, the captions would be preprocessed through the following steps:

1. First, all texts would be lower-cased.

2. Then, all punctuation would be removed and the texts would be tokenized (characters split into tokens, usually per word).[2]

3. Stopwords would then be removed, using the NLTK "english" stopword list. Stopwords are very common words, like "a", "the" etc.

[2] `http://www.nltk.org/_modules/nltk/tokenize/punkt.html#PunktLanguageVars.word_tokenize`

| Caption | Occurrences |
|---|---|
| fusion of multiple disc spaces squaring of the vertebral bodies fusion of si joints | 14 |
| extensive white matter lesions involving both cerebral hemispheres | 11 |
| fracture through the left c4 lateral mass and laminar arch with unilateral perched c45 facets on the right herniated and disrupted disk c45 torn intracapsular ligaments and ligament flavum | 11 |
| multilevel vertebral body lesions which are low signal on t1 and t2 scan sequences mediastinal adenopathy and perihilar nodular infiltrates on ct of chest | 11 |
| traumatic dislocation cervical spine at c1c2 level with marked widening of disc space and facet joints soft tissue edema anterior to spine and in posterior paraspinal locations edema and hemorrhage noted in lower medulla and upper cervical cord | 10 |

**Tab. 4.2:** The 5 most common captions of the ImageCLEF dataset.

| | |
|---|---|
| NORMAL | focal oval of fusiform activity along the medial aspect of the bilateral tibia |
| PREPROCESSED | focal oval fusiform activ along medial aspect bilater tibia |

**Tab. 4.3:** Example caption w/o (1st row) and w/ preprocessing (2nd row).

4. Finally, stemming would be applied, using the Snowball Stemmer from NLTK.[3]

Since the organizers would preprocess the submissions, there was no need for the submissions to be preprocessed beforehand, but also no harm. So should someone train their models on preprocessed text or not? Although these steps are common in preprocessing, a text may lose semantic value if preprocessed, or may be harder to understand, even though it is shorter (see Table 4.3). This means that models intended for real-life use should not generally be trained to predict preprocessed outputs, and scoring functions should punish this behavior. Early experiments showed that preprocessing the data beforehand was always resulting in better scores, thus the results shown later only consider models with preprocessed inputs.

## 4.2 IU X-Ray

The Indiana University Chest X-Ray Collection (IU X-Ray) is a public dataset of radiology images with their corresponding reports.[4] It contains 7,470 images and 3,851 human-written reports, one report for each patient (most patients have two images). The reports contain a variety of data. From those shown in Table 4.4, the caption was considered to be the Findings. Some examples can also be seen in Figure 4.3. For the splits, the ones used in

[3] `http://www.nltk.org/_modules/nltk/stem/snowball.html`
[4] Data can be found in: `https://www.kaggle.com/raddar/chest-xrays-indiana-university`. Original source: `https://openi.nlm.nih.gov/`

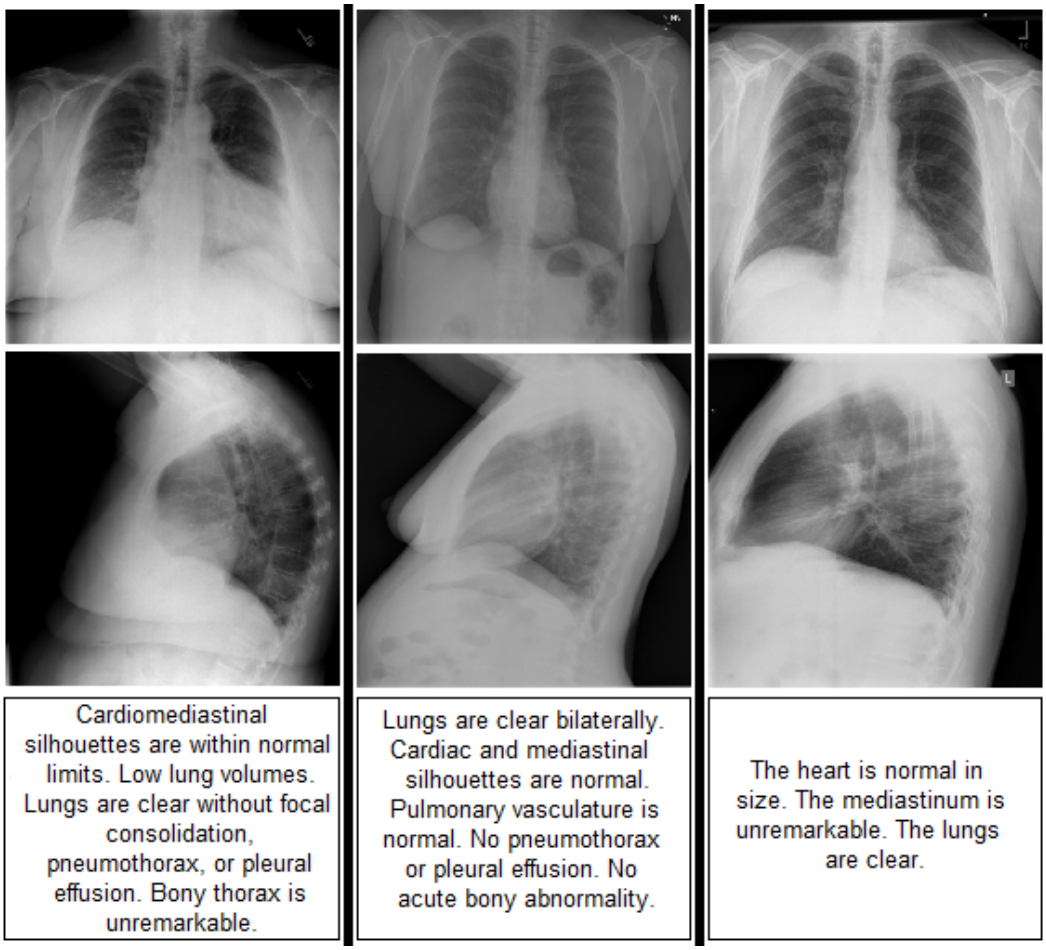


**Fig. 4.3:** Three random patients from the IU X-Ray dataset, along with their corresponding captions. Almost every patient has two images.

the R2Gen model were followed (70% for training, 10% for validation, and 20% for testing).[5] It was noticed that the total number of images was 5,910 instead of 7,470 and the total number of reports was 2,955 instead of 3,851, something that is not directly mentioned in the R2Gen paper [Che+20]. In the paper, it is mentioned that some images had to be dropped due to not having any reports. Like the previous dataset, IU X-Ray contains some repeated captions (407 out of the 2,365 captions of the training and validation sets, or about 17% of non-test captions), so retrieval methods can hope to retrieve identical reports. The most common captions can be seen in Figure 4.5. Two plots about the number of words in each caption, can be seen in Table 4.4.

As mentioned before, the current SOTA for this task is VisualGPT, while R2Gen is a strong competitor. But it was noticed that there is a difference between the way these models are scored. According to their repositories the preprocessing before scoring for VisualGPT involves removing punctuation, while the preprocessing before scoring for R2Gen removes all punctuation except full stops. This would not be a problem if the

[5] `https://github.com/cuhksz-nlp/R2Gen`

| | |
|---|---|
| **uid:** | 1 |
| **MeSH:** | normal |
| **Problems:** | normal |
| **image:** | Xray Chest PA and Lateral |
| **Indication:** | Positive TB test |
| **Comparison:** | None |
| **Findings:** | The cardiac silhouette and mediastinum size are within normal limits. There is no pulmonary edema. There is no focal consolidation. There are no XXXX of a pleural effusion. There is no evidence of pneumothorax. |
| **Impression:** | Normal chest x-XXXX |

**Tab. 4.4:** A report of the IU X-Ray dataset. The 7th row (Findings) was used as the Caption and everything else was dropped.

| Caption | Occurrences |
|---|---|
| The heart is normal in size. The mediastinum is unremarkable. The lungs are clear. | 35 |
| Heart size normal. Lungs are clear. XXXX are normal. No pneumonia, effusions, edema, pneumothorax, adenopathy, nodules or masses. | 28 |
| The heart and lungs have XXXX XXXX in the interval. Both lungs are clear and expanded. Heart and mediastinum normal. | 23 |
| The lungs are clear bilaterally. Specifically, no evidence of focal consolidation, pneumothorax, or pleural effusion.. Cardio mediastinal silhouette is unremarkable. Visualized osseous structures of the thorax are without acute abnormality. | 21 |
| Cardiac and mediastinal contours are within normal limits. The lungs are clear. Bony structures are intact. | 18 |

**Tab. 4.5:** The 5 most common captions of the dataset.

scoring functions ignored the full stops, but the scoring functions used in the R2Gen paper (that will be mentioned later) count full stops as tokens, if they are not removed by preprocessing beforehand. In Table 4.7, the difference between the captions can be observed, and although it may not seem important, later on it will be shown that there is a huge difference in the final scores sometimes.

If the differences in scores are ignored, a question is still left unanswered; should a model predict full stops? One could argue that outputs with full stops would be easier to interpret, but on the other side, someone else could state that they add little value to the meaning of a caption, and that expecting models to predict unnecessary tokens distracts them. In [Kou19] it can be seen that replacing full stops with an artificial token actually improved the scores of all their models. Also, full stops might actually hold significant semantic value. In Table 4.6 there's an example of a report (not a real one, but one that was created by the author), that shows how full stops could change the whole meaning of a caption.

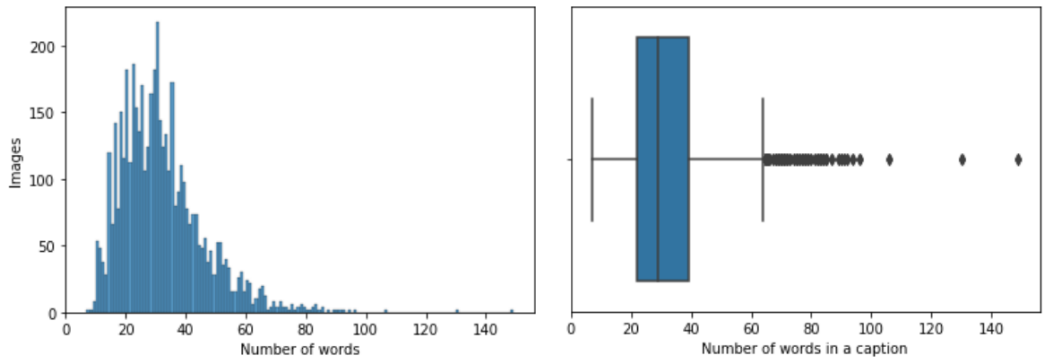


**Fig. 4.4:** Plots about the number of words in each caption, similar to the ones in Table 4.2. The data used was the training and validation sets. On the left, a histogram for the number of images with captions of a specific word length. On the right, a boxplot for the number of words inn captions.

| | |
|---|---|
| **falsely interpreted text** | about issues, the x-ray didn't show any lungs.<br>healthy heart not found.<br>damaged ribs are normal. |
| **correctly interpreted text** | about issues, the x-ray didn't show any.<br>lungs healthy.<br>heart not found damaged.<br>ribs are normal. |

**Tab. 4.6:** The difference between a text when the full stops change places. This means that without full stops, texts can be ambiguous. The text is not a real caption, it was written as an example.

The difference in VisualGPT and R2Gen does not involve the preprocessing before training, but rather the preprocessing after the prediction and before scoring. It was mentioned that full stops may help a model's training, but should they be considered as tokens when calculating scores, and if yes, how much should they affect the final scores? The only certainty is that there should be caution when comparing models from different papers, since it can be observed later on that the same predictions can have a big difference in scores, depending on the preprocessing used to the predictions before applying the score function.

| preprocess of: | sample |
|---|---|
| **None:** | The cardiomediastinal silhouette is within normal limits for size and contour. The lungs are normally inflated without evidence of focal airspace disease, pleural effusion, or pneumothorax. Osseous structures are within normal limits for patient age.. |
| **VisualGPT:** | the cardiomediastinal silhouette is within normal limits for size and contour the lungs are normally inflated without evidence of focal airspace disease pleural effusion or pneumothorax osseous structures are within normal limits for patient age |
| **R2Gen:** | the cardiomediastinal silhouette is within normal limits for size and contour . the lungs are normally inflated without evidence of focal airspace disease pleural effusion or pneumothorax . osseous structures are within normal limits for patient age . |

**Tab. 4.7:** A part of a caption (top), the desired output of VisualGPT (middle), and the desired output of R2Gen (bottom).

# 5 Results

The metrics used for captioning were BLEU [Pap+02] (in precise, its variants BLEU-1, BLEU-2, BLEU-3 and BLEU-4 were used), ROUGE-L [Lin04], and METEOR [LA07], while the accuracy of the models when used for predictive text will also be shown. The accuracy is calculated by the percentage of the correct next word predictions, while the definitions of the other metrics will not be provided, but some information will be given for these, as a general idea of how captioning models are scored.

1. **BLEU** uses the number of common n-grams (sequences of n tokens/words) among the predicted and gold text. Ignoring some details, BLEU-1 is a variant that is equal to the percentage of correct unigrams. BLEU-2 is a variant that is equal to the mean of the percentage of correct unigrams and 2-grams. BLEU-3 and BLEU-4 follow the same principle.

2. **METEOR** also uses unigrams. This score is based on how many tokens/words were guessed correctly, if they were put in correct order, and how far the positions of the correctly guessed tokens in the prediction were from the positions of the same tokens in the gold text.

3. **ROUGE-L** comes from a family of different score metrics, named ROUGE. Specifically, ROUGE-L takes into account the longest correctly guessed sequence of tokens and the lengths of the predicted and gold caption.

## 5.1 2021 ImageCLEFmedical results

First, the results in the 2021 ImageCLEFmedical Caption campaign will be shown. Table 5.1 shows all the models trained. Some were experimental runs on different pre-trained encoders (cp4 to cp13). As seen in the table, the image unaware language models (GPT-2 [Rad+19] and GPT Neo [Bla+21]) didn't outperform the retrieval models, which was expected since they were just simplistic baselines that ignore the images. And since the task is to generate diagnoses based on images, they should score very low, which is not the case; they even outperform the image aware encoder-decoder model used for this campaign.

| *ID* | Approach | BLEU-4 Score |
|---|---|---|
| cp1 | GPT-2 (117M parameters) | 34.923 |
| cp2 | GPT Neo (125M parameters) | 25.540 |
| cp3 | Show, Attend and Tell inspired | 20.471 |
| cp4 | DenseNet121 1-NN | 51.405 |
| cp5 | DenseNet201 1-NN | 52.755 |
| cp6 | ResNet50 1-NN | 52.256 |
| cp7 | ResNet152V2 1-NN | 42.120 |
| cp8 | InceptionV3 1-NN | 49.342 |
| cp9 | InceptionResNetV2 1-NN | 49.250 |
| cp10 | Xception 1-NN | 48.963 |
| cp11 | NASNetLarge 1-NN | 45.728 |
| cp12 | EfficientNetB0 1-NN | 51.747 |
| cp13 | EfficientNetB7 1-NN | 51.099 |
| cp14 | Tag-Trained ResNet50 1-NN | 50.988 |
| cp15 | Tag-Trained DenseNet201 1-NN | 53.381 |
| cp16 | Tag-Trained EfficientNetB0 1-NN | 52.641 |
| cp17 | Ensemble of cp5, cp8 and cp10 | 53.634 |
| cp18 | Ensemble of cp4, cp5, cp8, cp9 and cp10 | 54.153 |
| cp19 | Ensemble of cp4, cp5, cp8, cp9 and cp10 GPT-2 on non-Agreement | **55.342** |
| cp20 | Ensemble of cp14, cp15 and cp16 GPT-2 on non-Agreement | 54.877 |
| cp21 | Ensemble of cp5, cp6, cp14, cp15 and cp16 GPT-2 on non-Agreement | 55.023 |
| cp22 | cp19 with 2 most frequent sentences instead of most frequent caption | 51.161 |

**Tab. 5.1:** The scores of the group's systems in the validation set of the ImageCLEF dataset. The following was used to decide the submissions of the group. The cp1 and cp2 models are image unaware language models, and the model with ID cp3 is an encoder-decoder model. Models from cp4 to cp13 are 1-NNs with encoders pre-trained on ImageNet [Den+09], and models from cp14 to cp16 are 1-NNs with tag-trained encoders. The rest of the models involve ensembles of 1-NNs.

Two out of the three 1-NN models that used tag-trained encoders managed to score higher than the corresponding 1-NN models with the same encoder architectures pre-trained on ImageNet [Den+09], but the best model was an ensemble that used no 1-NNs with tag-trained encoder. Instead, the best model used five 1-NNs with different encoders pre-trained on ImageNet. For the best model, when all five 1-NNs disagreed on the output, the output of GPT-2 was used instead, being the best non-retrieval model. It can be seen in the scores table that ensembles outperformed the other models since the best ensemble scored about 3.67% higher (1.961 BLEU-4 difference) than the best non-ensemble model.

It was noticed from the obtained scores that some pre-trained encoders that have more parameters perform worse than similar pre-trained encoders with fewer parameters,

meaning that more complex architectures are not necessarily better. This observation is also shown in Table 5.2, where the similar architectures of encoders are grouped and the number of their parameters is given, along with their corresponding scores from Table 5.1 for their use in the 1-NN algorithm.

| *ID* | Encoder | Parameters | BLEU-4 Score |
|---|---|---|---|
| cp4 | DenseNet121 | 8.062.504 | 51.405 |
| cp5 | DenseNet201 | 20.242.984 | **52.755** |
| cp6 | ResNet50 | 25.636.712 | **52.256** |
| cp7 | ResNet152V2 | 60.380.648 | 42.120 |
| cp8 | InceptionV3 | 23.851.784 | **49.342** |
| cp9 | InceptionResNetV2 | 55.873.736 | 49.250 |
| cp12 | EfficientNetB0 | 5.330.571 | **51.747** |
| cp13 | EfficientNetB7 | 66.658.687 | 51.099 |

**Tab. 5.2:** A comparison of similar encoder architectures used for the 1-NN algorithm in the ImageCLEF dataset.

It should be noted that the 1-NN models do not have any hyperparameters to be tuned, which means that the training and validation sets can be merged. For the final submissions of 1-NN models, the whole dataset was used (training, validation and development combined). Six submissions were made by the group, which were based on the best models of the development set. The submissions can be seen in Table 5.3. It can be observed that the best model in the development set was also the best of the six submissions. The score differences between the development and the test set are big, but since most submissions involve 1-NNs with pre-trained encoders, it is not a problem of overfitting (because these models were not further trained). The best model was the 3rd ranked submission of the campaign. The first two ranked models of the campaign belonged to another team, but since they belonged to the same team, AUEB's NLP Group was the 2nd ranked team of the campaign.

| *ID* | BLEU-4 Score | | Rank |
|---|---|---|---|
| | Development | Test | |
| cp19 | **55.342** | **46.1** | **3** |
| cp21 | 55.023 | 45.2 | 4 |
| cp22 | 52.161 | 44.8 | 5 |
| cp17 | 53.634 | 44 | 7 |
| cp4 | 51.405 | 37.5 | 18 |
| cp3 | 20.471 | 19.9 | 38 |

**Tab. 5.3:** The final scores of the 6 submissions, along with their rank in the campaign.

## 5.2 IU X-Ray results

Regarding the results for the IU X-Ray models, Table 5.4 shows our experiments. It is observed that the R2Gen run of this thesis scored higher than the score mentioned in the R2Gen paper [Che+20], even though no changes to it were made, and that M2T was not performing well in the medical field. VisualGPT performed worse simply because some parameters, like batch size, had to be changed in order to train it (it needs high computational power otherwise). Also note that it was trained to maximize B-4, and it might achieve greater scores in the other measures if it was trained to maximize them instead. The 1-NNs use DenseNet201 as their encoder.

In the aforementioned table, *stacked* refers to the images of each patient; since most patients had two images, stacking them instead of giving one image per input was tested. *MatMul* refers to the use of matrix multiplication instead of cosine similarity (see more in Section 3.2). *R2Gen emb* refers to the use of the first representation layer of R2Gen for each image, instead of DenseNet201. *R2Gen embs* refers to the use of all the reprsentation layers of R2Gen for each image, instead of DenseNet201. *Uncertainty* is a name given by the author to the case of R2Gen having an average probability of the next words (across the whole prediction) lower than 0.955, a tuned number. The main idea is that when the model is not very certain of its prediction, another model should be used.

From the scores, we can see that the retrieval approaches are strong, as in the ImageCLEF campaign, but they couldn't compete with the scores of the SOTA models. It can also be observed that, when comparing two models, if one has a higher score in one metric, it can still have a lower score in another metric. An example of this is M2T and 1-NN, where M2T has about a 24.8% higher ROUGE-L score (difference of 6.3), but even if it scored a lot higher in that metric compared to 1-NN, it scored lower in METEOR.

Finally, regarding the difference between keeping the full stop or not, in the training and testing data. The best 1-NN (1-NN stacked) and R2Gen were tested in the 3 different preprocessing methods shown in Table 4.7 and got the results shown in Table 5.5. It can be noticed that the differences can be huge between the different kinds of preprocess.

| Model | B-1 | B-2 | B-3 | B-4 | ROUGE-L | METEOR | ACC |
|---|---|---|---|---|---|---|---|
| VisualGPT (on their paper) | **48.2** | **31.4** | 22.1 | 15.8 | **37.5** | **20.4** | - |
| R2Gen (on their paper) | 47.0 | 30.4 | 21.9 | 16.5 | 37.1 | 18.7 | - |
| VisualGPT (our run) | 30.6 | 19.2 | 13.3 | 9.4 | 30.8 | 13.7 | - |
| R2Gen (our run) | 47.8 | 31.3 | **22.8** | **17.5** | 36.3 | 19 | - |
| M2T | 34.0 | 21.0 | 15.0 | 10.9 | 31.7 | 15.1 | 6% |
| 1-NN | 33.6 | 19.5 | 12.5 | 8.3 | 25.4 | 15.3 | - |
| 1-NN stacked | 35.4 | 21.3 | 14.2 | 9.9 | 26.2 | 15.6 | - |
| 1-NN stacked MatMul | 32.5 | 18.9 | 12.0 | 7.9 | 24.7 | 14.9 | - |
| 1-NN stacked with R2Gen emb | 32.9 | 18.9 | 11.9 | 7.9 | 24.2 | 14.7 | - |
| 1-NN stacked with R2Gen embs | 33.9 | 20.0 | 12.9 | 8.7 | 25.6 | 15.3 | - |
| R2Gen + 1-NN on uncertainty | 34.0 | 21.4 | 15.5 | 12.0 | 27.2 | 17.7 | - |

**Tab. 5.4:** The final scores for IU X-Ray. The last column concerns using the models as a predictive text mechanism, which is not shown for 1-NN models since they do not use previous text as inputs. R2Gen and VisualGPT will be benchmarked for this task as future work. B-X is an abbreviation for BLEU-X. Please do note that R2Gen was not altered in any way, no credit is taken for its better performance than its paper.

| preprocess for | model | B-1 | B-2 | B-3 | B-4 | ROUGE-L | METEOR |
|---|---|---|---|---|---|---|---|
| **None** | $k$-NN | 26.7 | 15.8 | 10.5 | 7.5 | 20.1 | 16.8 |
| **None** | R2Gen | 25.2 | 14.3 | 8.8 | 5.7 | 19.8 | **18.9** |
| **VisualGPT** | $k$-NN | 35.4 | 21.3 | 14.2 | 9.9 | 26.2 | 15.6 |
| **VisualGPT** | R2Gen | 41.6 | 27.4 | 19.8 | 15.0 | 29.6 | 18.2 |
| **R2Gen** | $k$-NN | 41.9 | 25.4 | 17.1 | 12.2 | 31.7 | 16.7 |
| **R2Gen** | R2Gen | **47.0** | **30.8** | **22.4** | **17.2** | **36.3** | 18.7 |

**Tab. 5.5:** IU X-Ray scores, depending on the preprocess used for the gold captions (refer to Table 4.7).

# 6 Conclusions

## 6.1 Summary

AUEB's NLP Group managed to achieve the 2nd place at 2021 ImageCLEF's Captioning task, where the author was the main driver. More importantly, three varieties of models for diagnostic captioning were benchmarked; image unaware models, retrieval, and encoder-decoder models. Image unaware models were used as baselines. High scores in these baselines would be an indication that the captions of the dataset were very similar, so these models should not score very high, but surprisingly they outperformed the encoder-decoder model used in the ImageCLEF campaign. Retrieval approaches can be very effective in captioning, although they usually have difficulty combining and creating new captions, which means their outputs can fall short in variety. Encoder-decoder models are more complex, but they are the SOTA models for captioning in many captioning datasets, both biomedical and not. Even though there is research on them, they are not frequently tested in the biomedical domain, and the repository of the current SOTA of IU X-Ray (VisualGPT and R2Gen) needs time-consuming code changes and data handling in order to be executed for IU X-Ray. There was also an observation that the preprocessing of the captions can make a big difference between scores, so there should be standard rules about it when comparing models. On that observation, if the final outputs are preprocessed before given to the score funtion, then the model might make better scored predictions if it was trained to output preprocessed-like data, although they may be harder to interpret by humans.

## 6.2 Future Work

Since many captioning models available online have available code for the COCO dataset [Lin+14], and part of this thesis was using two of them in the biomedical domain, available code will be released to transform captioning datasets into a file format similar to that of COCO, to save time from these time-consuming steps of data transformation. Regarding future research, it would be interesting to test more models and datasets, also using captioning models that had not been tested on the biomedical domain. It would also be interesting to look more into how to combine different retrieved captions, and explore more visual extractors.

# Bibliography


[Aba+20] A. Ben Abacha, V. V. Datla, S. A. Hasan, D. Demner-Fushman, and H. Müller. "Overview of the VQA-Med Task at ImageCLEF 2020: Visual Question Answering and Generation in the Medical Domain". In: *CLEF (Working Notes)*. 2020.

[ACG20] K. C. Arnold, K. Chauncey, and K. Z. Gajos. "Predictive Text Encourages Predictable Writing". In: *Proceedings of the 25th International Conference on Intelligent User Interfaces*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 128–138.

[AKL19] A. Anastasopoulos, S. Kumar, and H. Liao. "Neural Language Modeling with Visual Features". In: (2019).

[All+17] M. Allahyari, S. Pouriyeh, M. Assefi, et al. "Text Summarization Techniques: A Brief Survey". In: *International Journal of Advanced Computer Science and Applications (ijacsa)* 8.10 (2017), p. 397.

[BCB15] D. Bahdanau, K. Cho, and Y. Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: abs/1409.0473 (2015).

[BKC17] V. Badrinarayanan, A. Kendall, and R. Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[Bla+21] S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman. *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow*. Version 1.0. Mar. 2021.

[Cha+19] S. Chaudhari, V. Mithal, G. Polatkan, and R. Ramanath. *An Attentive Survey of Attention Models*. Apr. 2019.

[Cha+21] F. Charalampakos, V. Karatzas, V. Kougia, J. Pavlopoulos, and I. Androutsopoulos. "AUEB NLP Group at ImageCLEFmed Caption Tasks 2021". In: *CLEF2021 Working Notes, CEUR Workshop Proceedings*. Bucharest, Romania, 2021.

[Che+20] Z. Chen, Y. Song, T. Chang, and x. Wan. "Generating Radiology Reports via Memory-driven Transformer". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Nov. 2020.

[Che+21] J. Chen, H. Guo, K. Yi, B. Li, and M. Elhoseiny. "Visualgpt: Data-efficient image captioning by balancing visual input and linguistic knowledge from pretraining". In: (2021).

[Cho+14a] K. Cho, B. van Merrienboer, Ç. Gülçehre, et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[Cho+14b] K. Cho, B. Van Merriënboer, C. Gulcehre, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: (Oct. 2014), pp. 1724–1734.

[Cho17] F. Chollet. *Deep learning with Python*. Simon and Schuster, 2017.

[Cor+20] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara. "Meshed-Memory Transformer for Image Captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[Den+09] J. Deng, W. Dong, R. Socher, et al. "ImageNet: A large-scale hierarchical image database". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Miami Beach, FL, USA, 2009, pp. 248–255.

[Dev+19] J. Devlin, M. Chang, K. Lee, and K. Toutanova. "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, 2019, pp. 4171–4186.

[Fis+20] A. Fisch, K. Lee, M. Chang, J. H. Clark, and R. Barzilay. "CapWAP: Captioning with a Purpose". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 8755–8768.

[HS97] S. Hochreiter and J. Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (1997), pp. 1735–1780.

[Hua+17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. "Densely Connected Convolutional Networks". In: (2017), pp. 4700–4708.

[Ion+21] B. Ionescu, H. Müller, R. Peteri, et al. "The 2021 ImageCLEF Benchmark: Multimedia Retrieval in Medical, Nature, Internet and Social Media Applications". In: *Lecture Notes in Computer Science* (2021).

[Kar+20] B. Karatzas, V. Kougia, J. Pavlopoulos, and I. Androutsopoulos. "AUEB NLP Group at ImageCLEFmed Caption 2020". In: *CLEF2020 Working Notes*. CEUR Workshop Proceedings. Thessaloniki, Greece, 2020.

[KB14] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations (ICLR)*. San Diego, CA, USA, 2014.

[Kou+21] V. Kougia, J. Pavlopoulos, P. Papapetrou, and M. Gordon. "RTEX: A novel framework for ranking, tagging, and explanatory diagnostic captioning of radiography exams". In: *Journal of the American Medical Informatics Association* (2021).

[Kou19] V. Kougia. "Medical Image Labeling and Report Generation". PhD thesis. Master Thesis-Athens University of Economics and Business (AUEB), Department of Informatics, 2019.

[KPA19a] V. Kougia, J. Pavlopoulos, and I. Androutsopoulos. "A Survey on Biomedical Image Captioning". In: *Workshop on Shortcomings in Vision and Language of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Minneapolis, MN, USA, 2019, pp. 26–36.

[KPA19b] V. Kougia, J. Pavlopoulos, and I. Androutsopoulos. "AUEB NLP Group at ImageCLEFmed Caption 2019". In: *CLEF2019 Working Notes*. CEUR Workshop Proceedings. Lugano, Switzerland, Sept. 2019.

[KPA20] V. Kougia, J. Pavlopoulos, and I. Androutsopoulos. "Medical Image Tagging by Deep Learning and Retrieval". In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020)*. Thessaloniki, Greece, 2020.

[KSZ14] R. Kiros, R. Salakhutdinov, and R. Zemel. "Multimodal Neural Language Models". In: vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 2014, pp. 595–603.

[LA07] A. Lavie and A. Agarwal. "METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments". In: *Proceedings of the second workshop on statistical machine translation*. 2007, pp. 228–231.

[LH18] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: (2018).

[Li+20] X. Li, X. Yin, C. Li, et al. "Oscar: Object-semantics aligned pre-training for vision-language tasks". In: *European Conference on Computer Vision*. Springer. 2020, pp. 121–137.

[Lin+14] T. Lin, M. Maire, S. Belongie, et al. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, pp. 740–755.

[Lin04] C. Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[Pap+02] K. Papineni, S. Roukos, T. Ward, and W. Zhu. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[Pav+21] J. Pavlopoulos, V. Kougia, I. Androutsopoulos, and D. Papamichail. "Diagnostic captioning: A Survey". In: (2021).

[PP20] J. Pavlopoulos and P. Papapetrou. *Clinical Predictive Keyboard using Statistical and Neural Language Modeling*. IEEE, 2020.

[Rad+19] A. Radford, J. Wu, R. Child, et al. "Language Models are Unsupervised Multitask Learners". In: vol. 1(8). 2019.

[Sin01] A. Singhal. "Modern information retrieval: A brief overview". In: *IEEE Data Eng. Bull.* 24.4 (2001), pp. 35–43.

[Sze+16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[Tak+20] E. Takmaz, S. Pezzelle, L. Beinborn, and R. Fernández. "Generating Image Descriptions via Sequential Cross-Modal Alignment Guided by Human Gaze". In: Association for Computational Linguistics. 2020.

[Vas+17] A. Vaswani, N. Shazeer, N. Parmar, et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[VKL20] H. Van, D. Kauchak, and G. Leroy. "AutoMeTS: The Autocomplete for Medical Text Simplification". In: *Proceedings of the 28th International Conference on Computational Linguistics*. 2020, pp. 1424–1434.

[Xu+15] K. Xu, J. Ba, R. Kiros, et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *Proceedings of the 32nd International Conference on Machine Learning*. Lille, France, 2015, pp. 2048–2057.

[Yua+19] Jianbo Yuan, Haofu Liao, Rui Luo, and Jiebo Luo. "Automatic radiology report generation based on multi-view image fusion and medical concept enrichment". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 721–729.