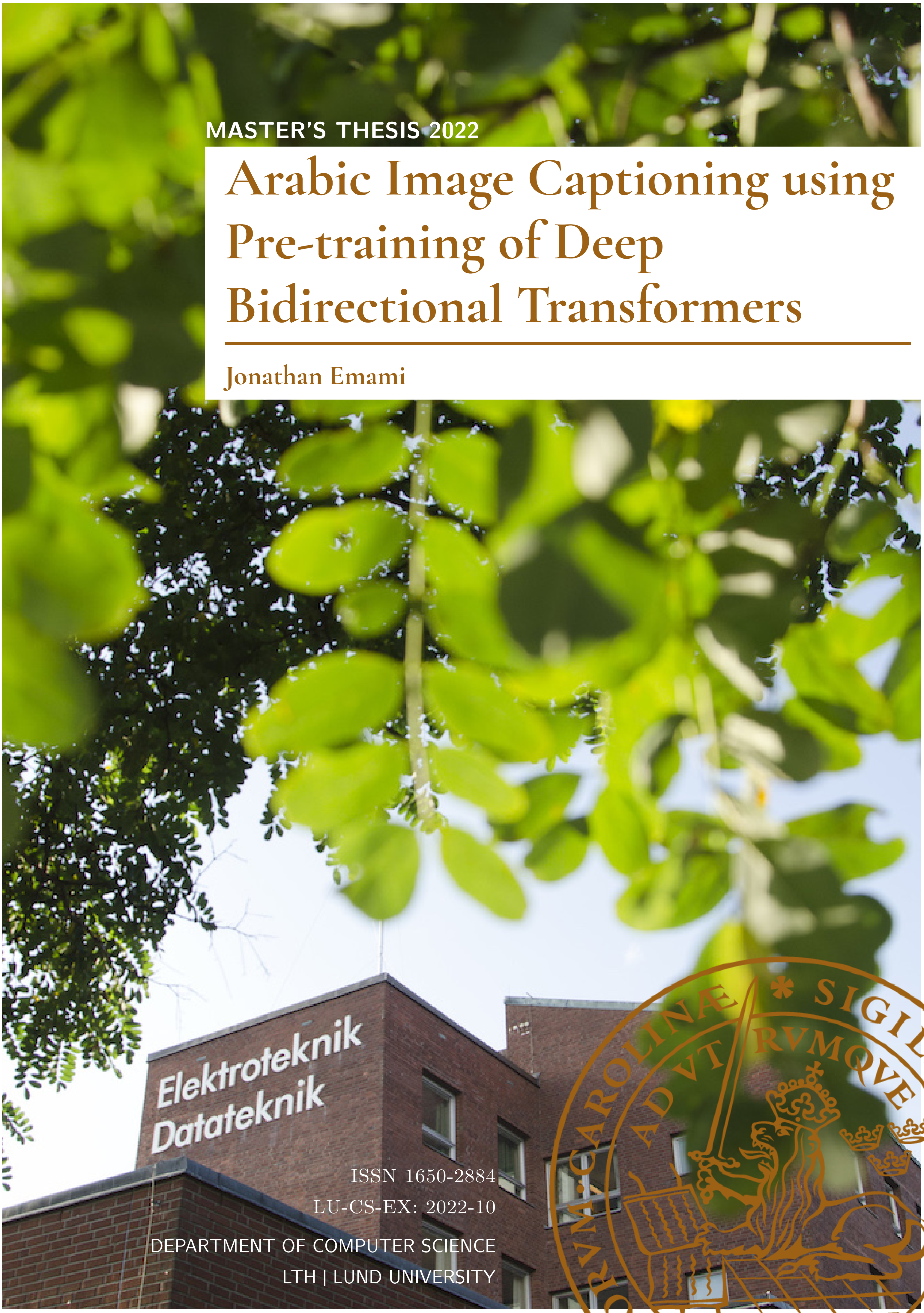
MASTER'S THESIS 2022

Arabic Image Captioning using Pre-training of Deep Bidirectional Transformers

Jonathan Emami



ISSN 1650-2884
LU-CS-EX: 2022-10
DEPARTMENT OF COMPUTER SCIENCE
LTH | LUND UNIVERSITY

EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2022-10

Arabic Image Captioning using Pre-training of Deep Bidirectional Transformers

Arabisk bildtextgenerering

Jonathan Emami

Arabic Image Captioning using Pre-training of Deep Bidirectional Transformers

Jonathan Emami
jontooy@gmail.com

March 17, 2022

Master's thesis work carried out at the Machine Learning and Arabic Language Processing Research Group, University of Sharjah.

Supervisors: Pierre Nugues, pierre.nugues@cs.lth.se
Ashraf Elnagar, ashraf@sharjah.ac.ae
Imad Afyouni, iafyouni@sharjah.ac.ae

Examiner: Jacek Malec, jacek.malec@cs.lth.se

Abstract

Image captioning is the process of automatically generating a textual description of an image. It has a wide range of applications, such as effective image search, auto archiving and even helping visually impaired people to see. English image captioning has seen a lot of development lately, while Arabic image captioning is lagging behind.

In this thesis, we developed and evaluated several Arabic image captioning models with well-established metrics on a public image captioning benchmark. We initialized all models with transformers pre-trained on different Arabic corpora. After initialization, we fine-tuned them with image-caption pairs using a learning method called OSCAR, which uses object tags detected in images as anchor points to significantly ease the learning of image-text semantic alignments.

In particular, we used AraBERT and GigaBERT as pre-trained models and performed training on two public datasets: One human verified and one machine translated. In relation to the image captioning benchmark, our best performing model scored 0.39, 0.25, 0.15 and 0.092 with BLEU-1,2,3,4 respectively, an improvement over previously published scores of 0.33, 0.19, 0.11 and 0.057. Beside additional evaluation metrics, we complemented our scores with a human evaluation on a sample of our output. Our experiments showed that training image captioning models with Arabic captions and English object tag labels is a working approach, but we conclude that a pure Arabic dataset, with Arabic labels, would be preferable.

Keywords: Arabic Image Captioning, Image Captioning, Transformers, Bert, Oscar, Pre-training, Vision and Language, Object Semantics

Acknowledgements

Thanks to Pierre Nugues for supervising and guiding my work. I would also like to thank Ashraf Elnagar and Imad Afyouni for valuable insights and proposing the subject of the thesis. Finally, thanks to my dad Hussain, for helping me with Arabic translations.

Contents

Chapter 1

Introduction

Vision-language tasks form a subfield of deep learning that unifies computer vision and natural language processing. Examples of vision-language tasks are visual question answering, image-text retrieval, and the most important for this study, image captioning.

Image captioning is the process of automatically generating a textual description of an image. It has a wide range of applications, such as effective image search, auto archiving and even helping visually impaired people to see. To illustrate, Figure 1.1 shows a machine generated caption on a picture of the University of Sharjah campus.

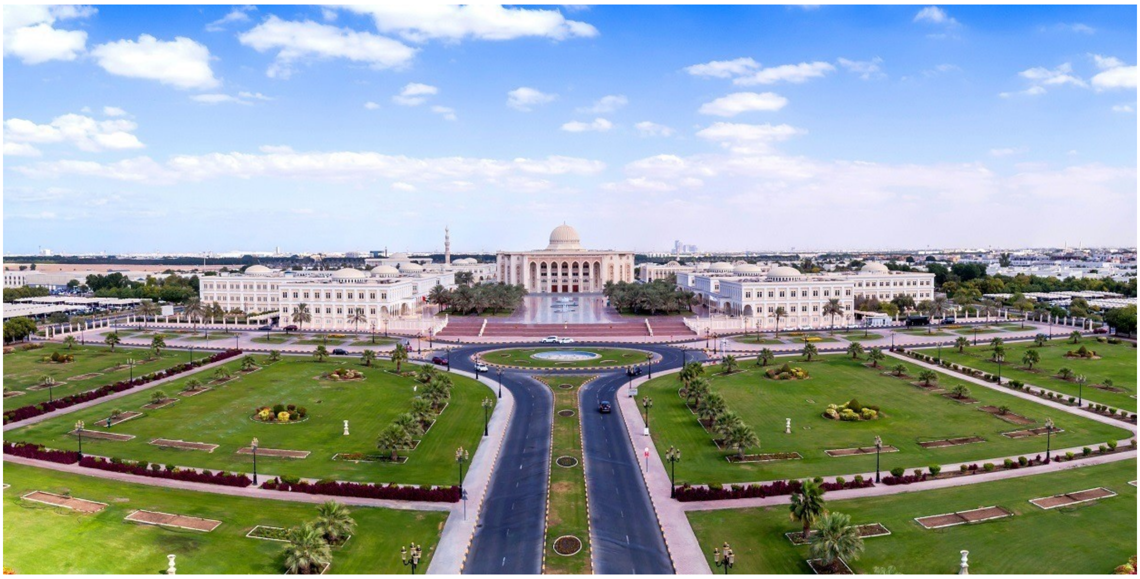


Figure 1.1: a large building with a park in front of it (machine-generated caption of a picture of the University of Sharjah).

State-of-the-art image captioning networks are today trained on English corpora and then translated to other languages, like Arabic. Arabic differs from the English language with a unique sentence structure, different spacing, and a very complex morphology. Because of these unique traits, machine translated captions become a source of error, and a step that should be eliminated for better results. A more attractive approach would be to train a model on an Arabic corpus from scratch, then fine tune it to fit appropriate evaluation metrics.

1.1 Background

The Machine Learning and Arabic Language Processing Research Group is a part of University of Sharjah, United Arab Emirates. Its aim is to develop local expertise and promoting awareness of the importance of Arabic language processing in the community at large.

The group's previous work on image captioning presents a hybrid solution to Arabic image captioning, which combines object detection and attention-based captioning techniques (Afyouni et al., 2021). All of their captioning models follow an encoder-decoder architecture. It consists first of a CNN image model to extract image features, then a language model, in their case a recurrent neural network (LSTM), to generate candidate captions. Their paper received the "Best paper award" from the International Conference on Artificial Intelligence in Computational Linguistics (ACLing 2021).

The research group has also released a paper survey on the current state of Arabic image captioning systems (Attai and Elnagar, 2020). In this survey, they conclude that the research conducted for Arabic image captioning is very scarce and that it can mainly be attributed to the lack of publicly available datasets. They also stress that few Arabic image captioning research projects utilized attention mechanisms, which is used to focus on the important parts of the image. Such attention mechanisms shall contribute to the caption generation process and give better results.

In their survey, Attai and Elnagar did not mention the transformer architecture as proposed by Vaswani et al. (2017), which is solely based on attention mechanisms. Moreover, transformers in natural language models are gaining more popularity as these models create new state-of-the-art results on different benchmarks, including Microsoft's English image captioning model OSCAR by Li et al. (2020).

1.2 Approach and goals

When we started this project, no transformer-based model for Arabic image captioning had been put to the test. The approach of this Master's thesis is to switch the language model of OSCAR with pre-trained Arabic and multilingual ones, then train them on Arabic benchmark datasets.

The comparison will be made by metrics used for evaluating automatic machine-translation software, like BLEU, ROUGE, and METEOR, but also image caption specific metrics, like CIDEr and SPICE. For comparisons of semantic meaning, we utilize the transformer-based Multilingual Universal Sentence Encoder (MUSE) and cosine similarity. Also qualitative assessments of the generated captions will be made.

As a summary, our goal is to evaluate transformer-based Arabic image captioning, by comparing our results to previous ones made by other researchers and create a roadmap for further research on this subject.

Chapter 2

Background

2.1 Related works

In this section, we will summarize the many recent developments in English image captioning. We will also comment on the current state of Arabic image captioning, and additional challenges that comes with Arabic language processing.

2.1.1 English Image Captioning

Attention is a technique in neural networks that mimics cognitive attention, and has shown great success in image captioning models ever since Xu et al. (2015) introduced an attention-based model that automatically learns to describe the contents of images. You et al. (2016) developed an algorithm that learns to selectively attend to semantic concept candidates and combine them with hidden states and outputs of recurrent neural networks. Huang et al. (2019) take the attention concept one step further in their work, where they propose an "Attention on Attention" (AoA) module, which extends the conventional attention mechanisms to determine the relevance between attention results and queries.

State-of-the-art image captioning today is based on transformers, an architecture that builds solely on attention mechanisms. Zhou et al. (2019) presented a unified vision-language pre-training (VLP) model which can be fine-tuned for both image captioning and visual question answering (VQA) tasks. Li et al. (2020) presented a new learning method OSCAR (**O**bject-**S**emanti**c**s **A**ligned P**r**e-training), and showed that learning of cross-modal representations can be significantly improved by introducing object tags detected in images. These object tags are used as "anchor points" during training to ease the learning of semantic alignments between images and texts. Zhang et al. (2021) studied improved visual representations, dubbed VinVL, and utilized an upgraded approach, dubbed OSCAR+, to pre-train transformer-based VL fusion models. They then fine-tuned the models on various VL benchmarks and created new state-of-the-art results on seven public benchmarks, including image captioning on the COCO Caption benchmark (see Section 2.2.1). In this Master's thesis, we utilized OSCAR with VinVL on Arabic image captioning.

2.1.2 Arabic Language Processing and Arabic Image Captioning

Arabic image captioning (AIC) introduces additional challenges compared to English captioning. In a paper survey on the state of AIC, Attai and Elnagar (2020) conclude that the research conducted for Arabic image captioning is very scarce and that it can mainly be attributed to the lack of publicly available datasets. The Arabic language is also known for its morphological complexity, and a variety of dialects, which makes it harder to process.

Arabic Language Processing. The Arabic language is extremely complex and therefore quite difficult to work with. The language has many different dialects and is mainly driven by the use of diacritics, a set of orthographic symbols that carry the intended pronunciation of words. For example, *harakat* (حَرَكَات) diacritics are used in the Qur'an but not in most written Arabic texts, to indicate short vowels, long consonants, and some other vocalizations.

Arabic also has a complex *morphology*. Morphology in linguistics deals with the structure of words and how they are formed. Sometimes words consist of solid stems (such as the Arabic noun يد (*yad*) "hand" or the English word "book"), but more often (especially in Arabic) words are composed of more than one morpheme. Examples of such are the English words *books*, *bookshelf*, *booked*; or the Arabic word مكتب (*maktab*) "office" consisting of the lexical root morpheme ك ت ب (K-T-B) "write" and the grammatical pattern morpheme specifying "place" (*ma __a _*). Some other words that can be formed using the root K-T-B are كِتاب (*kitab*) "book", كاتِب (*katib*) "writer", يَكتُبُ (*yaktubu*) "he writes", etc. A more complicated example would be words that could represent an entire sentence in English such as وسيحضرونها (*wasayahdurunaha*) "and they will bring it", which could be broken down into its morphemes و+س+ي+ حضر+ون+ها (*wa+sa+ya+hdr+runa+ha*) "and+will+bring+they+it".

Arabic language is also known for its lexical sparsity, which is due to the complex concatenative system of Arabic. For instance, the definite article ال (*Al*) is always prefixed to other words, although not an intrinsic part of the word. For example, كتاب (*kitab*) and الكتاب (*Alkitab*) are both included in the vocabulary, which leads to redundancy.

Arabic Image Captioning. Jindal leveraged the heavy influence of root-words to generate captions of an image directly in Arabic using root-word based recurrent neural networks (Jindal, 2017, 2018). They also reported the first BLEU score for direct Arabic caption generation, from experimental results on datasets from various Middle Eastern newspaper websites and the Flickr8k dataset (see Section 2.2.2).

Al-muzaini et al. (2018) developed a generative merge model for Arabic image captioning based on a deep RNN-LSTM and a CNN model. They used crowd sourcing to translate samples from two image captioning benchmarks: MS COCO and the Flickr8k dataset. They used a relatively small training set (2400 images) from an unpublished dataset. To reduce the risk of overfitting, ElJundi et al. (2020) developed an annotated dataset for Arabic image captioning (Flickr8k), which, as of today, remains to be the only public benchmark for AIC. They also developed a base model for AIC that relies on text translation from English image captions and compared it to an end-to-end model that directly transcribes images into Arabic text.

None of the works mentioned above utilized attention mechanisms in their proposed models. Afyouni et al. (2021) developed a hybrid object-based, attention-driven image captioning model. They performed a comprehensive set of experiments using popular metrics and multilingual semantic sentence similarity techniques to assess the lexical and semantic accuracy of generated captions.

Out of all the works from above, only ElJundi et al. (2020) have made their dataset publicly available, and is therefore the only work we can directly compare our models with.

When finishing this report, we discovered a Master's thesis, contemporaneous to our work by Sabri (2021). Though not a refereed publication, the author built neural network architectures which include techniques not previously explored in the Arabic image captioning literature, such as transformers. This approach yielded better results over the benchmark published by ElJundi et al. (2020).

2.2 Datasets

For this Master's thesis, we mainly used two public datasets for image captioning: Microsoft COCO and Flickr8k. We describe them in detail in the sections below. We also comment on other relevant datasets, such as WordNet and Visual Genome.

2.2.1 Microsoft COCO

Microsoft Common Objects in Context (COCO) (Lin et al., 2014) is a dataset consisting of 123,287 images including object detection, segmentation, and five captions per image (616,435 captions in total). As its name suggests, the COCO dataset contains complex everyday scenes with common objects in their natural context.

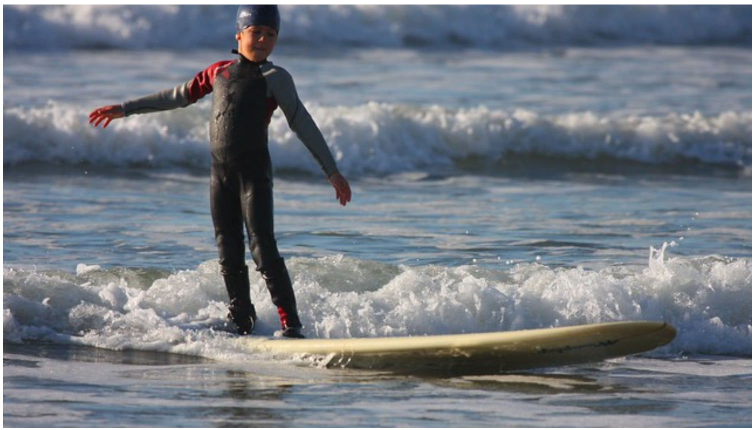
For comparison, we adopted the widely used Karpathy split of COCO (Karpathy and Fei-Fei, 2015), i.e. 113,287 train images, 5,000 validation images and 5,000 test images. The current best performing image captioning models on the Karpathy split can be found on the COCO image captioning online leaderboard[1].

We used 414,113 pre-translated captions over 82,783 training images with the Advanced Google Translate API[2], dubbed Arabic-COCO. Figure 2.1 shows an example of an image

[1] https://competitions.codalab.org/competitions/3221#results

[2] https://github.com/canesee-project/Arabic-COCO

from the train split with its five English captions and five Arabic captions. For the Arabic speaking reader, note the error in the machine translated caption nr. 2, where the phrase “ ركوب الأمواج”, should be replaced with its present tense “ويركب الموج”.



#	English captions	Arabic captions
1.	A young boy surfing in low waves.	صبي صغير يتزلج على الأمواج المنخفضة.
2.	A young boy is standing on a surfboard and riding a wave.	صبي صغير يقف على لوح ركوب الأمواج وركوب الأمواج.
3.	A surfer rides his surf board on some very small waves.	راكب أمواج يركب لوح الأمواج على بعض الأمواج الصغيرة جدًا.
4.	A young boy is standing on a surfboard in the water.	صبي صغير يقف على لوح تزلج على الماء في الماء.
5.	A young boy is standing on a surfboard in the ocean.	صبي صغير يقف على لوح ركوب الأمواج في المحيط.

Figure 2.1: Caption annotations in English and Arabic for an image sample from the COCO dataset.

2.2.2 Flickr8k

The Flickr8k dataset (Hodosh et al., 2013) consists of 8,092 images. Each image in this dataset is associated with five different captions that describe the entities and events depicted in the image. They were collected via a crowdsourcing marketplace (Amazon Mechanical Turk) with a total of 40,460 captions.

Human translations into Arabic of both the COCO and Flickr8k datasets have been done before. For example, Al-muzaini et al. (2018) built an Arabic dataset based on these two English benchmark datasets. Most of them are not public, therefore we used Arabic Flickr8k by ElJundi et al. (2020). Arabic Flickr8k is split into 6,000 train images, 1,000 validation images, and 1,000 test images, all with three Arabic captions each.

The translation to Arabic was performed by ElJundi et al. in two steps, first by using the Google Translate API and then by validating captions with professional Arabic translators. Finally, they chose the top three translated captions out of five for each image, which makes 24,000 captions in total. Figure 2.2 shows an example of an image from the train split with its three original English captions and three verified Arabic captions. Note that even though verified, the quality of these Arabic captions is questionable. For example caption 2 in Figure

2.2 says “رجل أسود”, which incorrectly translates to “black man”.

#	English captions	Arabic captions
1.	A longhaired man surfing a large wave.	رجل طويل الشعر يتزلج موجة كبيرة
2.	A man in black on a surfboard riding a wave.	رجل أسود على لوح ركوب الأمواج يركب موجة
3.	A man surfing in the ocean.	رجل يمارس رياضة ركوب الأمواج في المحيط

Figure 2.2: Caption annotations in English and Arabic for an image sample from the Flickr8k dataset.

The Flickr30k dataset (Young et al., 2014) consists, as the name implies, of over 30,000 (31,783 to be exact) photographs of everyday activities, events, and scenes with five captions each, i.e. 158,915 captions in total. It contains and extends the work by Hodosh et al. (2013) and follows the same annotation, guidelines and quality controls. It is widely used in image captioning research, but was not used in this project, due to the lack of public Arabic translated captions.

Table 2.1 shows the complete list of image caption datasets used in this report.

Datasets	Train		Validation		Test	
	#Images	#Captions	#Images	#Captions	#Images	#Captions
Arabic-COCO	82, 783	414, 113	-	-	-	-
Flickr8k	6, 000	18, 000	1, 000	3, 000	1, 000	3, 000
TOTAL	88, 783	432, 113	1, 000	3, 000	1, 000	3, 000

Table 2.1: Statistics for the Arabic-COCO and Flickr8k translated by ElJundi et al. (2020).

2.2.3 WordNet and Visual Genome

WordNet. WordNet was originally created as a lexical database for the English language (Miller, 1995). The database contains 155,327 words organized in 175,979 *synsets*. Synonymy is WordNet’s basic relation, and WordNet uses sets of synonyms (synsets) to represent word senses. For example, `man` would get mapped to

`man.n.03 (the generic use of the word to refer to any human being)`

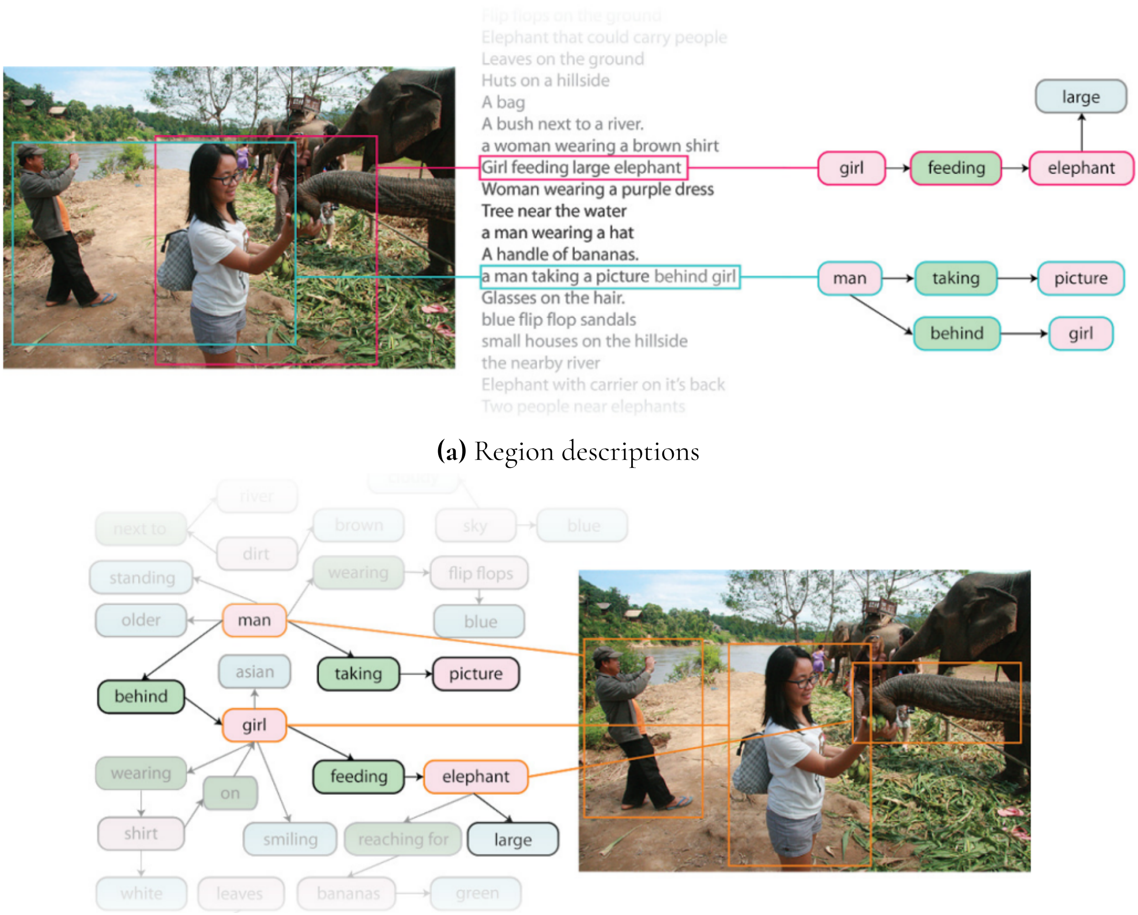
Similarly, `person` gets mapped to:

```
person.n.01 (a human being)
```

Afterwards, these two concepts can be connected since `person.n.01` is a hypernym of `man.n.03`. WordNet has since its creation been expanded to linked to over 200 languages, including Arabic.

Visual Genome. Visual Genome is a dataset created to connect structured image concepts to language (Krishna et al., 2017). The authors represent objects, attributes, relationships, and noun phrases with region descriptions, and connect them with scene graphs, see Figure 2.3. All of the objects, attributes, and relationships in each image in the Visual Genome dataset are canonicalized, i.e. uniquely mapped, to their corresponding WordNet ID (synset ID). This mapping connects all of the images in Visual Genome and provides an effective way to consistently query the same concept in the dataset. In addition, their dataset also contains image-related question-answer pairs.

We used label maps provided by Visual Genome to label detected objects from this project. We used the same label map as Zhang et al. (2021), which includes 1,594 object labels.



(a) Region descriptions

(b) Scene graph

Figure 2.3: A sample image from the Visual Genome dataset. (a) shows examples of region descriptions (e.g. "girl feeding large elephant" and "a man taking a picture behind girl"). (b) shows the objects (e.g. elephant), attributes (e.g. large) and relationships (e.g. feeding) connected in a scene graph. After Krishna et al. (2017).

Chapter 3

Theory

3.1 Encoder-decoder architectures for image captioning

There are two general paradigms in existing image captioning approaches: top-down and bottom-up. The top-down paradigm starts from a "gist" of an image and converts it into words, while the bottom-up one first comes up with words describing various aspects of an image and then combines them. Language models are employed in both paradigms to form coherent sentences.

The state of the art today follows the top-down paradigm, and has an end-to-end formulation from an image to a sentence, based on a encoder-decoder architecture:

1. Creating a gist from an image through a convolutional neural network (CNN) encoder, a procedure known as *image feature extraction*.

2. Generating a sentence from the gist through a language model, for example through a recurrent neural network (RNN) decoder.

The idea of this formulation is that all the parameters of the recurrent network can be learned from training data. This formulation is in principle similar to the deep neural network (DNN) architecture introduced by Sutskever et al. (2014) to solve the English-to-French machine translation problem. Instead of English-to-French translation, we are looking for an image-to-language translator, where, in our case, the language is Arabic.

Instead of using recurrent neural networks, we will use *transformers* with the OSCAR learning method proposed by Li et al. (2020). As we will see in later sections, we can use a transformer model for both encoding words and decoding the final image caption.

3.2 Image feature extraction and object tag detection

The first step of most image caption generators is to extract features from the image. This enables us to later on train our model to map these image features and sentence features into a common space, knows as *alignment*. Image-text alignment may be used for grounding natural language symbols to the physical world and semantically understanding the content of an image.

As an example model for feature extraction, Zhang et al. (2021) trained a large-scale object and attribute detection model based on the ResNeXt-152 C4 architecture (Xie et al., 2016), short as X152-C4. ResNeXt is named after and adopts the ResNet strategy, a residual learning framework designed to ease the training of networks that are substantially deeper than those used previously (He et al., 2016). For this Master's thesis, we will utilize X152-C4 for feature extraction, pre-trained on 2.49 million unique images, including the COCO and VG dataset. Figure 3.1 shows an example of object detection with the X152-C4 model.

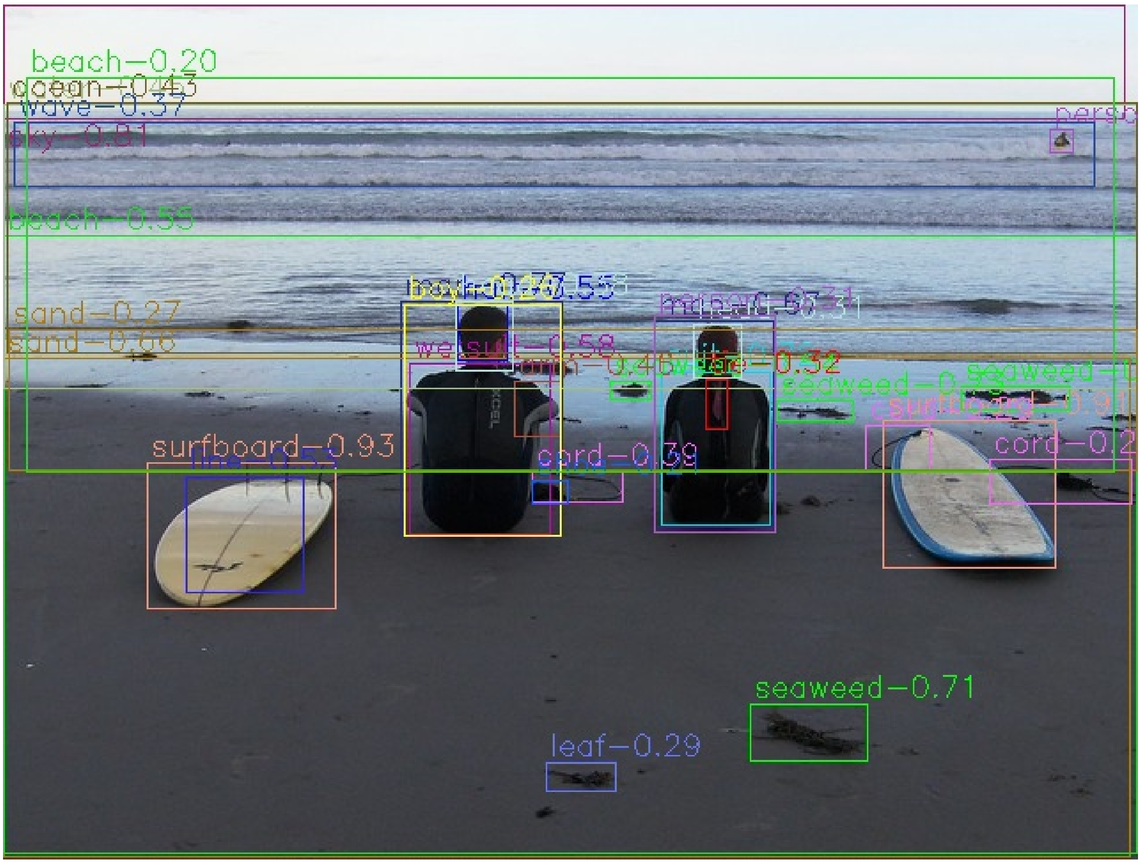


Figure 3.1: Object detection on an image from the COCO dataset using the X152-C4 architecture. The set of detected object labels are `(Arm, Beach, Boy, Cord, Hair, Head, Leaf, Line, Man, Ocean, Person, Sand, Seaweed, Sky, Suit, Surfboard, Tie, Water, Wave, Wetsuit)`.

For each detected object, an image region vector $\boldsymbol{v}$ is generated. Each region feature $\boldsymbol{v}$ is denoted as $(\hat{v}, z)$, where $\hat{v}$ is the 2048-dimensional vector input to the last linear classification layer and z is a 6-dimensional position encoding of the region. More specifically

$$z = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \Delta x \\ \Delta y \end{bmatrix},$$

where the points (x_1, y_1) and (x_2, y_2) define a region bounding box. Both the x and y axes are scaled down with the image width and height respectively. The detected image region vectors are finally vertically stacked and saved. $\boldsymbol{q}$ is the set of detected object labels outputted by the object detector for each image.

3.3 Tokenizers

Before processing text to the models, the raw text needs to be *tokenized*, i.e. broken down into small chunks. For example, the sentence "It is raining" can be tokenized in many ways. Using white space tokenization, the sentence can be broken down into words "It", "is" and "raining", while subword tokenization can break down the sentence even further, breaking down the word "raining" into its components "rain" and "##ing". Here the the ## indicates that the followed token belongs to the previous one and that they are one word in the input sentence. The main advantage of subword tokenization is that it interpolates between word-based and character-based tokenization, which makes it very useful for specific applications where the subwords make sense. In the following paragraph, we will shortly describe and comment different kinds of subword tokenization algorithms.

3.3.1 Byte-Pair Encoding

Sennrich et al. (2015) adapted the byte-pair encoding (BPE) (Gage, 1994) algorithm to tokenize raw text into words and subwords. Instead of merging frequent pairs of bytes, they merge characters or character sequences. Firstly, they initialize a token vocabulary size. Simplified, the following steps involved in BPE algorithm are given below:

1. Start with splitting the raw input text into single unicode characters. Each of the characters corresponds to a symbol in the final vocabulary.

2. Find the most frequent occurring pair of symbols from the current vocabulary and add this to the vocabulary.

3. Repeat step 2 till the defined token vocabulary size is reached, or the highest symbol frequency is one.

3.3.2 WordPiece

Schuster and Nakajima (2012) introduced the WordPiece algorithm when trying to solve the Japanese and Korean voice problem. The algorithm is comparable to BPE, with a slight difference in the method of choosing the subwords. BPE greedily considers the token with most frequent occurring pair of symbols to merge into the vocabulary, while WordPiece considers the frequency of individual symbols before merging into the vocabulary. More specifically, WordPiece merges the pair of symbols that maximizes the likelihood of the training data. Maximizing the likelihood of the training data is equivalent to finding the symbol pair (x, y), whose probability divided by the probabilities of its first symbol followed by its second symbol is the greatest among all symbol pairs, i.e. choose symbols (x, y) s.t.

$$\underset{(x,y)}{\operatorname{argmax}} \frac{P(x, y)}{P(x)P(y)}$$

The WordPiece algorithm gained popularity through the famous state-of-the-art model BERT, which we will discuss in later sections.

3.3.3 SentencePiece

The tokenization algorithms discussed above requires that the input text is separated by a space between the words. In most languages this is often the case, apart from some languages like Chinese and Japanese. Kudo and Richardson (2018) created the SentencePiece algorithm, which does not use space as a separator. Instead, it takes the input as a string in its original raw format, i.e. together with all the spaces, and then uses BPE or other tokenizers to create the vocabulary.

3.4 Word Embeddings

In order to represent image features and words in the same vector space, we need to vectorize each word in our corpus, i.e. map each word to a vector.

One possible vectorization technique is to use *embeddings*. Embeddings are dense vector representations of words from 10 to a few hundreds dimensions. In addition, most embedding techniques allow words with similar meanings to be close in the vector space (Mikolov et al., 2013a). Moreover, some embeddings have compositional properties. For example, Mikolov et al. (2013b) explain how the male/female relationship is automatically learned by utilizing induced vector representations, and that the vector addition “King - Man + Woman” results in a vector very close to “Queen”.

GloVe is a popular word embedding used today (Pennington et al., 2014), but one problem with this model is that it does not take the word context into account. For example, consider the sentence “I must go back to my ship and to my crew”. The word “ship” can be a verb or a noun with different meanings, but has only one GloVe embedding vector. As we will see in the next section, transformer-based word embeddings solve this problem with self-attention, which enables *contextual word embeddings*. In this Master’s thesis, all the word embeddings were obtained from transformer-based models.

3.5 The Transformer

The transformer architecture builds solely on attention mechanisms and was first proposed by Vaswani et al. (2017). The transformer has shown great success in sequence-to-sequence modeling, and the key to success lies in the possibility to train semantic relations on very large corpora and memorize them in matrices.

In the paper *Attention is all you need*, Vaswani et al. (2017) used three kinds of vectors: queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$, and values $\boldsymbol{V}$. The attention vector is then computed in the following way:

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

where d_k is the dimension of the input and the softmax function is defined as

$$\text{softmax}(x_1, x_2, ..., x_n) = \left(\frac{e^{x_1}}{\sum_{i=1}^{n} e^{x_i}}, \frac{e^{x_2}}{\sum_{i=1}^{n} e^{x_i}}, ..., \frac{e^{x_n}}{\sum_{i=1}^{n} e^{x_i}}\right)$$

For in-depth explanations of the original transformer architecture, the reader is referred to the original article by Vaswani et al. (2017).

3.5.1 BERT

Proposed by Devlin et al. (2019), BERT is short for Bidirectional Encoder Representations from Transformers. When released, Devlin et al. (2019) showed that pre-trained representations reduce the need for many heavily-engineered task-specific architectures. In other words, by pre-training general language representations, BERT was the first fine-tuning based representation model that achieved state-of-the-art performance on a large group of sentence-level tasks, outperforming many task-specific architectures.

For tokenization, they used WordPiece (Schuster and Nakajima, 2012) with a 30,000 token vocabulary in English. What makes BERT unique is its two training objectives, masked language modeling (MLM) and next sentence prediction (NSP):

MLM: A technique that randomly masks a portion of the input tokens (words), then aims to predict the masked tokens. In BERT's case, 15% of all WordPiece tokens in each sequence were masked and predicted during pre-training.

NSP: This technique allows the BERT model to understand the relationship between sentences by simply training the model, when given two sentences *A* and *B*, to predict (with a 50% chance) whether *B* is the sentence following *A* or not.

During pre-training, they fed the model two sentences at the time. They used three added special tokens: `[CLS]` at the start of the first sentence as a special classification token, `[SEP]` to separate both sentences and the token `[MASK]` to denote the words to predict.

With their paper, Devlin et al. (2019) released two pre-trained models BERT$_{\text{BASE}}$ and BERT$_{\text{LARGE}}$, with the latter containing more parameters and layers than the former. The release of BERT preceded many other BERT-based language models trained on different corpora from different languages, and will be the main base for our image captioning model. The following paragraphs describe the models used in this Master's thesis. All of them were initialized on the BERT$_{\text{BASE}}$-configuration. Table 3.1 shows the different models configurations for comparison.

mBERT. mBert, short for Multilingual BERT, was pre-trained with the multilingual Wikipedia dataset that consists of the top 104 most common languages (Devlin et al., 2018), including Arabic. In this comparison, we used the `bert-base-multilingual-uncased` version of mBERT from HuggingFace. This is a version of BERT that does not differentiate between capitalized and noncapitalized words. For instance, the words "Arabic" and "arabic" are considered the same in this version of BERT. Note that there is no capital letters in the Arabic written language, which makes the distinction useless.

AraBERT. AraBERT was released by Antoun et al. (2020) and was among other models compared to mBert. AraBERT achieved state-of-the-art performance on most tested Arabic NLP tasks. The models were trained on news articles manually scraped from Arabic news websites and several publicly available large Arabic corpora. One of the corpora is named OSCAR (**O**pen **S**uper-large **C**rawled **A**ggregated Co**r**pus), not to be confused with the image captioning model OSCAR (**O**bject-**S**emanti**c**s **A**ligned P**r**e-training). In total, the dataset consists of ~77GB of text. There are several versions of AraBERT available. We used the `bert-base-arabertv02` configuration in this project.

ArabicBERT. ArabicBERT (Safaya et al., 2020) was the first pre-trained BERT model for Arabic at the time. It was originally pre-trained as an approach to solve a sub-task of the Multilingual Offensive Language Identification shared task (OffensEval 2020), which is a part of the SemEval 2020, a series of international NLP research workshops. The training dataset consists of a dump of Arabic Wikipedia and an Arabic version of OSCAR, summing up to ~95GB of text in total. We used the `bert-base-arabic` configuration in this project.

GigaBERT. GigaBERT (Lan et al., 2020) is a set of models pre-trained as a bilingual BERT and designed specifically for Arabic NLP and English-to-Arabic zero-shot transfer learning. Their best model significantly outperforms mBERT and AraBERT on some supervised and zero-shot transfer settings. The training dataset consists of a dump of Arabic Wikipedia, an Arabic version of OSCAR and the Gigaword corpus, which consists of over 13 million news articles. We used the `GigaBERT-v4-Arabic-and-English` configuration in this project.

Models	**Training Data**		**Vocabulary**			**Configuration**	
	source	#tokens (all/ar)	tokenization	size (all/ar)	cased	size	#parameters
mBERT	Wiki	21.9B/153M	WordPiece	110k/5k	no	base	172M
AraBERT	Wiki, Oscar, News articles	2.5B/2.5B	SentencePiece	64k/58k	no	base	136M
ArabicBERT	Wiki, Oscar	unknown	WordPiece	32k/28k	no	base	111M
GigaBERTv4	Wiki, Oscar, Gigaword	10.4B/4.3B	WordPiece	50k/26k	no	base	125M

Table 3.1: Configuration comparisons for mBert, AraBERT, ArabicBERT, and GigaBERT.

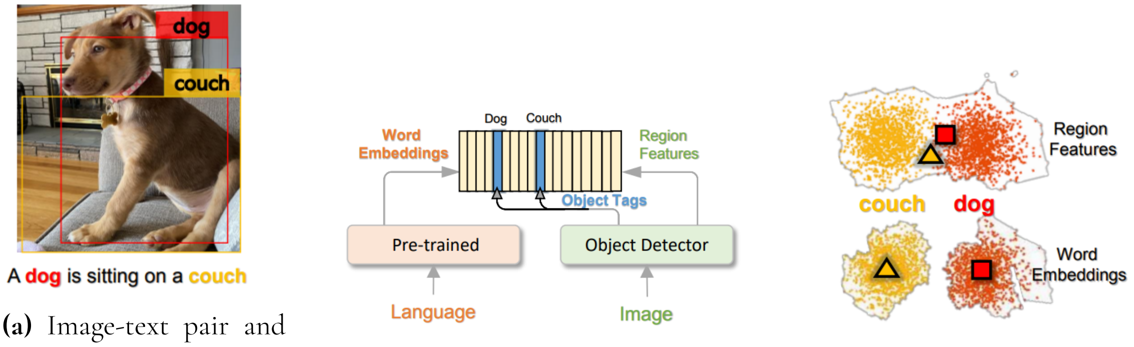
3.6 Vision-Language Pre-training

The vanilla $\text{BERT}_{\text{BASE}}$ cannot handle image region features as input. As a learning method, we used OSCAR (Li et al., 2020), which achieves state-of-the-art results on six well-established vision-language understanding and generation tasks, including image captioning. Previous pre-training methods concatenate image region features and text features as input and then use self-attention to learn image-text semantics in a brute force manner.

OSCAR uses object tags detected in images as anchor points to ease the alignment of image region and word embeddings. The method is motivated by the observation that the salient objects in an image can be accurately detected by modern object detectors and that these objects are often mentioned in the caption. Except for a novel input representation, we will also describe the pre-training objective used in the original OSCAR model, and the fine-tuning objective used in this Master's thesis.

3.6.1 Input Representation

OSCAR represents each input image-text pair as a Word-Tag-Image triple $(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v})$. Here $\boldsymbol{w}$ represent the caption, while $\boldsymbol{q}$ and $\boldsymbol{v}$ represent the objected object tags and object region features as described in Section 3.2. The idea is that the alignments between $\boldsymbol{q}$ and $\boldsymbol{w}$, both in text, are relatively easy to identify by using pre-trained BERT models, which are used as initializations in Oscar (Li et al., 2020). The task then becomes to train the model to ground the image objects in distinctive entities represented in the language space. See Figure 3.2 as an example for visual and textual representations of a dog sitting on a couch.



(a) Image-text pair and object tags

(b) Input representation

(c) Semantic spaces

Figure 3.2: (a) Word-Tag-Image triple with tags colored in red and yellow. (b) Input vector design. Note how the caption is processed through a pre-trained BERT model, while the image is processed though an object detector and then concatenated with the word embedding and the tags as anchor points. (b) The word semantic space is more representative than image region features. In this example, "dog" and "couch" are similar in the visual feature space due to the overlap regions, but distinctive in the word embedding space. This is often the case, due to visual regions often being over-sampled, noisy and ambiguous. After Li et al. (2020).

In the context of Arabic image captions, this leaves us with a couple of options. To feed Arabic captions to the OSCAR model, one could either:

1. Use Arabic captions $\boldsymbol{w}$ and English labels $\boldsymbol{q}_{\text{Eng}}$ generated by the X152-C4 object detector. The OSCAR model is then trained on a multilingual BERT, for example mBert or GigaBERT.
2. Use Arabic captions $\boldsymbol{w}$ and Arabic labels $\boldsymbol{q}_{\text{Arab}}$, either obtained from
   (a) $\boldsymbol{q}_{\text{Eng}}$ being mapped to $\boldsymbol{q}_{\text{Arab}}$ through WordNet or, as the case of this project,
   (b) $\boldsymbol{q}_{\text{Eng}}$ being directly machine translated to $\boldsymbol{q}_{\text{Arab}}$, for example through the Google Translate API.

   The OSCAR model is then trained on an Arabic BERT, for example ArabicBERT or AraBERT.

3.6.2 Pre-training Objective

In the orginal OSCAR paper (Li et al., 2020), the model is pre-trained from two different perspectives, named the *dictionary view* and the *modality view*:

Dictionary view: Similar to the masked token loss (MTL) used by BERT, they define the discrete token sequence as $\boldsymbol{h} = [\boldsymbol{w}, \boldsymbol{q}]$, and apply the MTL for pre-training. At each iteration, they randomly mask each input token in $\boldsymbol{h}$ with probability 15%, and replace the masked one $\boldsymbol{h}_i$ with a special token [MASK]. The goal of training is to predict these masked tokens based on their surrounding tokens $\boldsymbol{h}_{\setminus i}$ and all image features $\boldsymbol{v}$ by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{MTL}} = -\mathbb{E}_{(\boldsymbol{v},\boldsymbol{h})\sim\mathcal{D}} \log p(h_i|\boldsymbol{h}_{\setminus i}, \boldsymbol{v})$$

Modality view: Utilizes a contrastive loss. For each input triple, they group $\boldsymbol{h}' = [\boldsymbol{q}, \boldsymbol{v}]$. They then sample a set of polluted image representations by replacing $\boldsymbol{q}$ with probability 50% with a different tag sequence randomly sampled from the dataset $\mathcal{D}$. Since the encoder output on the special token [CLS] is the fused vision-language representation of $(\boldsymbol{h}', \boldsymbol{w})$, they apply a fully-connected (FC) layer on top of it as a binary classifier $f(.)$ to predict whether the pair contains the original image representation $(y = 1)$ or any polluted ones $(y = 0)$. The contrastive loss is defined as

$$\mathcal{L}_{\text{C}} = -\mathbb{E}_{(\boldsymbol{h}',\boldsymbol{w})\sim\mathcal{D}} \log p(y|f(\boldsymbol{h}', \boldsymbol{w}))$$

The full pre-training objective of OSCAR is then simply the sum of these losses:

$$\mathcal{L}_{\text{Pre-training}} = \mathcal{L}_{\text{MTL}} + \mathcal{L}_{\text{C}}$$

For more details about the training objective implementations and in-depth explanations, please read the original paper (Li et al., 2020).

Zhang et al. (2021) introduced an improved pre-training objective of OSCAR, called OSCAR+, were they instead of the binary constrastive loss above apply a novel 3-way contrastive loss $\mathcal{L}_{\text{CL3}}$, with the purpose of effectively optimizing the training objectives used for vision question answering (VQA) and text-image matching. Since we are only interested in image captioning, we will not describe the details of this training objective.

3.6.3 Fine-tuning objective

The original OSCAR adapts the pre-trained models to seven downstream VL tasks, each one posing different challenges for adaption. Since we are only concerned with image captioning, we will focus on the image captioning fine-tuning strategy.

According to the recipe described by Li et al. (2020), the input samples are processed to Word-Tag-Image triples $(\boldsymbol{w}, \boldsymbol{q}, \boldsymbol{v})$ in the same way as that during the pre-training. 15% of the caption tokens are then randomly masked out, and the remaining context of the triple is used to predict the masked tokens with a cross-entropy loss. Since the BERT-based model is bidirectional, the self-attention mask during training is constrained such that a caption token can only attend to the tokens before its position to simulate a unidirectional generation process. Note that all of the caption tokens will have full attentions to image regions and object tags but not the other way around.

Zhang et al. (2021) used the Karpathy split on a pre-trained OSCAR+$_{\text{BASE}}$, then fine-tuned with cross-entropy loss for 30 epochs, a batch size of 256 and an initial learning rate of $1e^{-5}$. Finally, they used a so-called self critical sequence training (SCST) (Rennie et al., 2017) to optimize the CIDEr evaluation metric (see Section 3.8) for 10 epochs, a batch size of 128 and an initial learning rate of $2e^{-6}$. In this Master's thesis, we did not use a pre-trained OSCAR. Instead, our model was directly initialized from a BERT$_{\text{BASE}}$ configuration, and then trained on our caption data with cross-entropy loss, no CIDEr optimization used.

3.7 Image Captioning Inference

We used the caption inference procedure described by Li et al. (2020). During inference, they first encode the image regions, object tags, and a `[CLS]` token as input. They then initialize the caption generation by feeding in a `[MASK]` token and sampling a token from the vocabulary based on the likelihood of the output. Next, the `[MASK]` token in the previous input sequence is replaced with the sampled token and a new `[MASK]` is appended for the next word prediction. The generation process terminates when the model outputs the `[STOP]` token.

In the greedy decoding of candidate captions, we consider a single token at every step. With a *beam search* decoder, we could track multiple tokens at every step and use those to generate multiple candidate sentences, then pick the candidate sentence that maximizes the log likelihood. Beam search is an algorithm that uses breadth-first search to build a search tree. Li et al. (2020) used beam search with a beam size of 5, i.e. they expanded the search tree to the top 5 contenders after every token prediction.

3.8 Evaluation metrics

There are many ways of evaluating image captions. Viewing image captioning as a image-to-text machine translation, we could use classical machine translation metrics, such as BLEU, ROUGE, and METEOR, to measure the quality of our generated captions. We could also use other well-established caption evaluation metrics, such as CIDEr and SPICE. Since our dataset contains several ground-truth captions per image, another approach would be to use sentence embeddings, for example MUSE, and find the ground-truth caption closest to our generated caption in the semantic space. Since most of the current evaluation metrics are built for the English language, it is appropriate to complement the scores with a qualitative assessment made by native Arabic speakers. In the following section, we will introduce the evaluation metrics used in this project and comment on them.

3.8.1 BLEU

Papineni et al. (2002) proposed BLEU (short for **Bil**ingual **E**valuation **U**nderstudy) as a method for the automatic evaluation of machine translation that is quick and language-independent. Following previous works, we evaluated our captioning models on the BLEU-1,2,3,4, which assesses a candidate sentence (generated caption) by measuring the fraction of n-grams that appear in a set of references (ground-truth captions). More specifically, an individual BLEU n-score is calculated as the *modified precision* of a candidate sentence. To compute precision, they simply count the number of candidate translation words which occur in any reference translation and then divide by the total number of words in the candidate translation. For example, consider the following reference and candidate sentence pair:

- Reference: `The cat is on the mat.`
- Candidate: `the the the the the the the.`

The important words for computing modified precision are here underlined. The modified unigram precision is 2/7, since a reference word is considered exhausted after a matching candidate is identified. The nature of the precision metric makes a perfectly translated BLEU score 1 and a perfect mismatch 0. Note that even human translators do not achieve a perfect score of 1.

In this project, we made use of all of the individual n-gram scores, which capture two aspects of translation: *adequacy* and *fluency*. A translation using the same words (1-grams) as in the references tends to satisfy adequacy. The longer n-gram matches (2-, 3- and even 4-grams) account for fluency (Papineni et al., 2002). It is also important to note that the more reference translations per sentence there are, the higher the score is.

Trying to compare BLEU scores across different corpora and languages is strongly discouraged. One important observation is that Arabic candidate captions tend to score lower on BLEU-scores compared to their English counter parts. To demonstrate this, consider the following reference and candidate sentence pair in Arabic and English:

- Reference: `He walked on the beach at night` – تمشى على الشاطئ ليلاً
- Candidate: `She walked on the beach at night` – تمشت على الشاطئ ليلاً

The differing words between the two sentences are underlined. The two sentences give respective BLEU-1,2,3,4 scores 0.75, 0.71, 0.63, 0.00 in Arabic, but higher scores 0.86, 0.85, 0.83, 0.81 in English. This observation is caused by the morphological complexity of the Arabic language, which leads to Arabic sentences being shorter and therefore making error penalties much higher in n-gram based metrics like BLEU.

3.8.2 ROUGE

Lin (2004) introduced a package, ROUGE (Recall-Oriented Understudy for Gisting Evaluation), for automatic evaluation of summaries and its evaluations. It includes several automatic evaluation methods that measure the similarity between summaries, including ROUGE-L, that was used in this project.

ROUGE-L is based on an F-measure. The traditional F-measure is calculated as the harmonic mean of *precision* and *recall*. A more general F-score, F_β, that uses a positive real factor β, where β is chosen such that recall is considered β times as important as precision, is:

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}, \tag{3.1}$$

where P and R are the calculated precision and recall, respectively.

ROUGE-L deals with the longest common sub-sequences (LCS) between two summaries (captions) X and Y. Lin (2004) uses a LCS-based F-measure to estimate the similarity between two summaries X of length m and Y of length n, assuming X is a reference summary sentence (ground-truth captions) and Y is a candidate summary sentence (candidate caption). The F-measure F_{lcs} (ROUGE-L score) is calculated according to Equation 3.1, with P_{lcs} as our subsequence precision (see Equation 3.2) and R_{lcs} as our subsequence recall (see Equation 3.3):

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{3.2}$$

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{3.3}$$

$LCS(X,Y)$ is the length of a longest common subsequence of X and Y. Notice that ROUGE-L is 1 when $X = Y$, while ROUGE-L is 0 when $LCS(X,Y) = 0$, i.e. there is nothing in common between X and Y ((Lin, 2004)). To illustrate how this measure differs from previous ones, consider the following reference sentence and candidate sentences:

- Reference: `Police killed the gunman`
- Candidate 1: `Police kill the gunman`
- Candidate 2: `the gunman kill police`

BLEU-2 would score both candidates the same, since they share the same bigram, i.e. "the gunman". However, Candidate 1 and Candidate 2 have very different meanings. In the case of ROUGE-L, Candidate 1 has a score of 3/4 = 0.75 and Candidate 2 has a score of 2/4 = 0.5, with β = 1. One other obvious difference between ROUGE-L and BLEU-n is that since it automatically includes longest in-sequence common n-grams, you don't need a predefined n-gram length.

3.8.3 METEOR

METEOR (**M**etric for **E**valuation of **T**ranslation with **E**xplicit **O**rde**r**ing) was proposed by Banerjee and Lavie (2005) and was designed to explicitly address several observed weaknesses in the BLEU metric. In their paper, the authors demonstrate that METEOR significantly improves correlation with human judgments and show that recall plays a more important role than precision in obtaining high-levels of correlation with human judgments. METEOR is based on unigram-precision and recall (Banerjee and Lavie, 2005).

Similar to ROUGE-L, METEOR calculates an F-measure for each reference and candidate pair. The F-measure is computed as follows: First the unigram precision (P) is computed as the ratio of the number of unigrams in the candidate sentence that are *mapped* (to unigrams in the reference sentence) to the total number of unigrams in the candidate sentence. Similarly, the unigram recall (R) is computed as the ratio of the number of unigrams in the candidate sentence that are mapped (to unigrams in the reference sentence) to the total number of unigrams in the reference sentence. Next, they compute F_{mean} by combining the precision and recall via a harmonic mean, see Equation 3.1, that places three times as much importance on recall than on precision (i.e. $\beta = 3$).

METEOR finally computes a penalty for a score, having the effect of reducing the F_{mean} to a maximum of 50% if there are no bigram or longer matches. Notice that METEOR always scores in the interval $[0, 1]$. For more details about the mapping process and penalty calculation, read the original paper (Banerjee and Lavie, 2005).

3.8.4 CIDEr

CIDEr (Consensus-based Image Description Evaluation) was developed by Vedantam et al. (2014) specifically for image caption evaluation, and measures the similarity of a candidate sentence to the majority, or *consensus*, of a set of ground truth sentences written by humans.

The CIDEr score calculation is more complicated than the previous F-measures, and we will briefly describe it in this paragraph. All of the words in the sentences (both candidate and references) are first mapped to their stem or root forms. That is, "fishes", "fishing" and "fished" all get reduced to "fish", or as previously mentioned in Section 2.1.2, in Arabic كِتاب (*kitab*) "book", كاتِب (*katib*) "writer", يَكتُبُ (*yaktubu*) "he writes" all get reduced to the root morpheme ك ت ب (K-T-B). The stemmed sentences are then represented using the set of n-grams present in it. In their paper, they use n-grams containing one to four words. n-grams that commonly occur across all of the images in the dataset should be given lower weight, since they are likely to be less informative. This observation is encoded through *term frequency-inverse document frequency* (tf-idf) weighting for each n-gram Vedantam et al. (2014).

The tf-idf is calculated as the product of the term-frequency (tf) and the inverse document frequency (idf). The term frequency is defined as

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, \tag{3.4}$$

where $f_{t,d}$ is the raw count of a term (word or token) t in a document d (candidate or reference sentence). The inverse document frequency is defined as

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}, \tag{3.5}$$

where N is the total number of documents in the corpus ($N = |D|$) and $|\{d \in D : t \in d\}|$ is number of documents where the term t appears. Finally, tf-idf is calculated as a multiplication of Equation 3.4 and 3.5, see Equation 3.6:

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D). \tag{3.6}$$

Intuitively, tf places higher weight on n-grams that frequently occur in the reference sentence, while idf reduces the weight of n-grams that commonly occur across all of the images in the dataset.

The $\mathrm{CIDEr}_n(c_i, r_i)$ score for n-grams of length n is computed using the average cosine similarity between the tf-idf weighting of the candidate sentence c_i and the reference sentences r_i. The definition of the *cosine similarity* $S_{\cos\theta}$ between two vectors $\boldsymbol{v}$ and $\boldsymbol{u}$ should be familiar to the reader, see Equation 3.7:

$$S_{\cos\theta} = \frac{\boldsymbol{v} \cdot \boldsymbol{u}}{||\boldsymbol{v}||\,||\boldsymbol{u}||}. \tag{3.7}$$

Generally, this similarity ranges to any value in the interval $[-1, 1]$, but since the tf-idf weighting is non-negative, our CIDEr_n scores will be in the interval $[0, 1]$. The CIDEr_n scores for every n-gram are finally combined by averaging:

$$\mathrm{CIDEr}(c_i, r_i) = \frac{1}{4} \sum_{n=1}^{N} \mathrm{CIDEr}_n(c_i, r_i).$$

The metric used in this report is a modified version of CIDEr called CIDEr-D (Vedantam et al., 2014). In this new formula, the authors propose the removal of stemming. Since singular and plural forms of nouns and different tenses of verbs are being mapped to the same token, the removal of stemming ensures the correct forms of words are used. To reduce *gameability* of the metric, the authors also penalize scores based on the difference between candidate and reference sentence lengths and repetition of confident words or phrases until the desired sentence length is reached.

A factor of 10 is added to the calculation of the CIDEr-D scores numerically to make them similar to other metrics. One consequence of this factor is that CIDEr-D values often exceeds 1, and in fact have the maximum value of 10.

3.8.5 SPICE

SPICE (Semantic Propositional Image Caption Evaluation) is a metric developed by Anderson et al. (2016), as the name suggest, for image captioning. SPICE differs from previous scores in that it does not utilize n-gram overlaps, which the authors mean are neither necessary, nor sufficient for two sentences to convey the same meaning. The authors mention two examples sentences:

1. `A young girl standing on top of a tennis court.`
2. `A giraffe standing on top of a green field.`

The captions describe two very different images. However, comparing these captions using any of the previously mentioned n-gram metrics produces a high similarity score. To overcome this problem, the authors estimate caption quality by transforming both candidate and reference captions into a graph-based semantic representation, i.e. a *scene graph*. To complete this task, they adopt the Stanford Scene Graph Parser followed by post-processing steps, including resolving pronouns and handling plural nouns (Anderson et al., 2016). Finally an F-score is calculated over logical tuples representing semantic propositions in the generated scene graph.

To exemplify, we revisit example sentence 1. First, the sentence is parsed into a dependency parse tree, see Figure 3.3. A dependency parse tree is a graph that represents the syntactic structure of a sentence according to grammatical dependency relations.

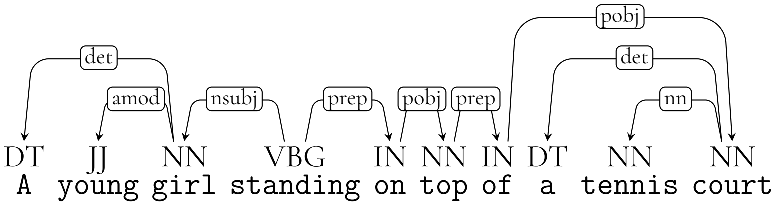

pobj
det
det
amod
nsubj
prep
pobj
prep
nn
DT JJ NN VBG IN NN IN DT NN NN
A young girl standing on top of a tennis court

Figure 3.3: Example sentence parsed into a dependency parse tree.
After Anderson et al. (2016)

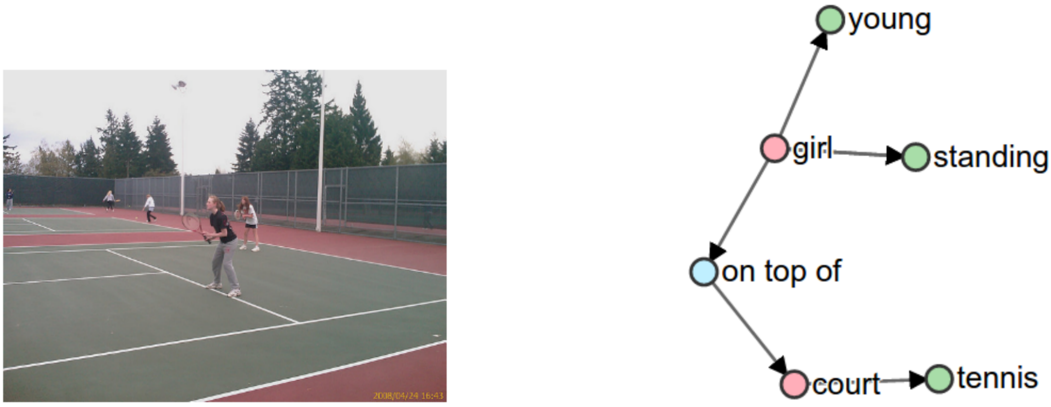
The dependency tree is then mapped into a scene graph, see Figure 3.4.

From the scene graph, Anderson et al. (2016) extract a set of tuples, each containing either one, two or three elements, representing objects, attributes and relations, respectively. The example in Figure 3.4 would be represented with the following tuples:

{ (girl), (court), (girl, young), (girl, standing), (court, tennis), (girl, on-top-of, court) }

Viewing the semantic propositions of a candidate sentence as a set of tuples, the same tuples can be matched against a set of tuples obtained from the scene graph of a set of reference sentences. From this matching, a precision and recall score can be calculated and finally combined into the SPICE F-score.

Being an F-score, SPICE is simple to understand, and naturally bounded to the interval $[0, 1]$. Unlike CIDEr, SPICE does not use cross-dataset statistics, and is therefore equally applicable to both small and large datasets.



(a) Example image

(b) Scene graph

Figure 3.4: (a) Image corresponding to the example sentence. (b) A visualization of a semantic scene graph with encoded objects (e.g. girl), attributes (e.g. young) and relations (e.g. on top of) present. After Anderson et al. (2016).

A potential concern that the authors address is that the metric could be gamed by generating captions that represent only objects, attributes and relations, while ignoring other important aspects of grammar and syntax. Because SPICE neglects fluency, it is implicitly assuming that captions are well-formed (Anderson et al., 2016).

In the context of Arabic image caption evaluation, the SPICE metric described in Anderson et al. (2016) can not be directly applied. For Arabic, as for other morphologically rich languages, the role of morphology is often expected to be essential in syntactic modeling, and the role of word order is less important than in morphologically poorer languages such as English. Notwithstanding the use of the Stanford Scene Graph Parser, their proposed SPICE metric is not tied to this particular parsing pipeline, and could potentially be replaced with an Arabic one.

3.8.6 MUSE

MUSE (Multilingual Universal Sentence Encoder) is a multilingual sentence embedding model released by Yang et al. (2020). The model embeds text from 16 languages, see Table 3.2, into a shared semantic space. The model achieved new state-of-the-art performance on several NLP tasks, such as monolingual and cross-lingual semantic retrieval tasks.

In this report, although we used Arabic-specific BERT models for caption generation, we used MUSE embeddings to compute similarity scores between generated captions, as sentence embeddings proved to perform better on multilingual semantic textual similarity tasks. This technique is similar to the one developed by Afyouni et al. (2021).

Although there is an initial intensive computational requirement for the sentence embeddings, the formula for the similarity score is simple. Firstly, the cosine similarity $S_{\cos\theta}$ between two vectors $\boldsymbol{v}$ and $\boldsymbol{u}$ is calculated according to Equation 3.7. From the resulting scalar, an *angular distance* D_θ is calculated, see Equation 3.8. Finally, the *angular similarity* S_θ between two vector embeddings $\boldsymbol{v}$ and $\boldsymbol{u}$ (see Equation 3.9) is simply the complement of the

Languages	**Family**
Arabic	Semitic
Chinese, Chinese (Taiwan)	Sino-Tibetan
Dutch, English, German	Germanic
French, Italian, Portuguese, Spanish	Latin
Japanese	Japonic
Korean	Koreanic
Russian, Polish	Slavic
Thai	Kra-Dai
Turkish	Turkic

Table 3.2: Multilingual universal sentence encoder's supported languages. After Yang et al. (2020).

angular distance D_θ:

$$D_\theta = \arccos(S_{\cos\theta})/\pi = \frac{\theta}{\pi} \tag{3.8}$$

$$S_\theta = 1 - D_\theta = 1 - \frac{\theta}{\pi} \tag{3.9}$$

When similarity scores between candidate captions and all their respective reference sentences have been calculated, score statistics and qualitative assessments can be made.

3.8.7 Human Evaluation

Since some evaluation metrics, like BLEU and SPICE discussed above, are not optimized for Arabic captions, human evaluation has to be made to verify the quality of the candidate captions. In this thesis, we chose to focus on the qualities of *grammar*, *semantics*, and *context*. For this task, native Arab speaking experts evaluated a sample of the candidate captions generated across the proposed models.

There are several ways of evaluating image captions manually. In this thesis, we followed the guidelines of the Transparent Human Benchmark (THUMB), a human evaluation protocol proposed by Kasai et al. (2021). Like previously described automatic scores, the authors base their evaluations on two main scores (precision and recall) and three types of penalties (*fluency*, *conciseness*, and *inclusive language*). The overall score is computed by averaging precision and recall and deducting penalty points.

Precision (P) measures how precise the caption is given the image, while Recall (R) measures how much of the salient information (e.g., objects, attributes, and relations) from the image is covered by the caption. Both scores are assessed in the scale of 1–5.

Kasai et al. (2021) found most captions from modern neural network models were highly fluent and concise. Since precision and recall covers the context of an image, our penalty will be purely based on grammar and semantics.

Chapter 4
Evaluation

4.1 Experimental setup

We initialized the captioning model with various Arabic-specific BERT configurations. In order to select the best models, we carried out two experiments considering the multi/bilingual aspects and the learning curve of the fitting procedure:

1. Evaluation of two multilingual models both trained on

   (a) Arabic captions and Arabic labels

   (b) Arabic captions and English labels

   We carried out this experiment mainly for comparing the object labels ability to affect the final image-text alignment.

2. Evaluation of the learning curve for three different models, respectively trained on 50%, 75% and 100% of a dataset. From the results we can tell if the validation loss decreases with the amount of data or if some adjustment have to be made to the models, for example with a hyper parameter grid search. Out of the trained models, we chose the two most accurate ones as candidates for large scale training.

After we picked two candidate models, a third and final experiment was made:

3. Do large scale training on the candidate models on datasets of different size. Evaluate the models both with automatic and human metrics and compare the results with previous models.

We carried out the first two experiments on Google Colab GPU:s (1 P100 GPU with 16 GB memory). We carried out the final large scale experiments on a workstation (1 GV100 GPU with 32 GB memory) and a high performance computer (HPC) system (8 K80 GPU:s with 12 GB memory each), both provided by the University of Sharjah.

4.1.1 Preprocessing

Before training the models, we ran all of the images through the X152-C4 object detector for extraction of region features and object tags. Since all of the image features and object tag labels are made available for the Karpathy split of the COCO dataset by Li et al. (2020), only Flickr8k images had to be inferred. We then split the Flickr8k image features and object tags into train, validation and test images following ElJundi et al. (2020).

To train models on Arabic captions and Arabic labels, we simply translated English labels directly with the Google Translate API, as described in Section 3.6.1.

4.1.2 Training and evaluation

In the first experiment, we initialized our multilingual models with GigaBERT (GigaBERT-v4-Arabic-and-English) and mBERT (bert-base-multilingual-uncased). We trained both models twice for 30 epochs with a learning rate of $1e^{-4}$ on the Flickr8k train-split and validated on the Flickr8k val-split, with Arabic and English labels respectively. After training, we applied an image caption inference on the val-split for every saved model checkpoint. We finally ran the BLEU-1, 2, 3, 4, ROUGE-L, METEOR, CIDEr and SPICE evaluation scripts on the inferred candidate captions.

In the second experiment, we initialized our model with AraBERT (bert-base-arabertv02), ArabicBERT (bert-base-arabic) and GigaBERT (GigaBERT-v4-Arabic-and-English). We trained the three models for 30 epochs with Arabic labels, a learning rate of $1e^{-4}$ on 50%, 75% and 100% of the Flickr8k train-split captions. For the AraBERT and GigaBERT configurations specifically, the learning rate η was grid searched in the interval $[1e^{-5}, 7e^{-5}]$ with 100% of the Flickr8k data to find the best configuration. We validated all of the models on the Flickr8k val-split during training time. Post training, we made image caption inference on the val-split for every saved model checkpoint. We chose to only run the MUSE evaluation script on the inferred candidate captions, since we are most interested in how well the different models learned object semantics. Also, CIDEr is not equally applicable on both small and large datasets (see Section 3.8). Since our models are trained on 50%, 75% and 100% of the original dataset, CIDEr could not be used for model comparison.

In the third and last experiment, we picked two candidate models: AraBERT (bert-base-arabertv02) and GigaBERT (GigaBERT-v4-Arabic-and-English). For each candidate model, we trained 3 captioning models on 3 different datasets: the Flickr8k train-split, Arabic-COCO, and then a mix of Arabic-COCO and the Flickr8k train-split (88,783 images and 432,113 different captions in total). We trained the models for 30 epochs when trained on the Flickr8k train-split, and 50 epochs when trained on the Arabic-COCO and the mixed dataset. All of the experiments were repeated with batch sizes 32 and 265 respectively. Post training, we applied the image caption inference on the Flickr8k test-split for the last checkpoint on every saved model. We finally ran the BLEU-1,2,3,4, ROUGE-L, METEOR, CIDEr, SPICE and MUSE evaluation scripts on the inferred candidate captions.

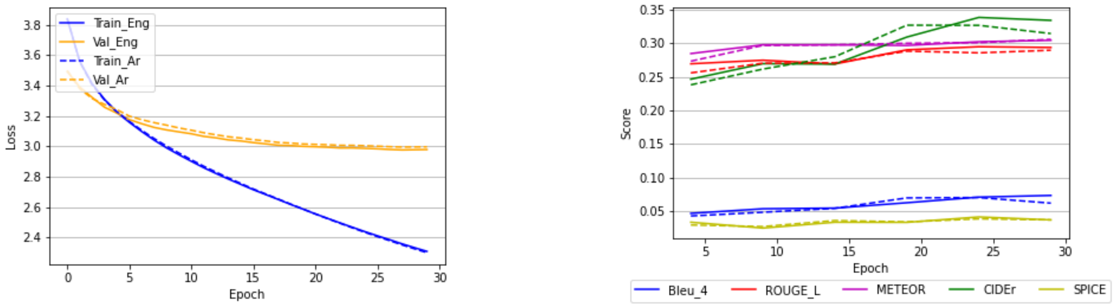
For all of the experiments above, we saved training and validation loss values at every epoch, while model checkpoints were saved every 5 epochs. All of the experiments used the AdamW opptimizer and a linearly decaying learning rate according to the recipe described in OSCAR (Li et al., 2020). Exact model hyper parameters for each experiment are shown in the Appendix A section.

4.2 Results

We inferred all of the candidate captions through beam search, with a width of 5 and a max sentence length of 20. After inference, we calculated evaluation scores (BLEU-1,2,3,4, ROUGE-L, METEOR, CIDEr and SPICE) using the COCO Caption evaluation API[3]. For MUSE, we used universal-sentence-encoder-multilingual-large version 3, downloaded from TensorFlow Hub[4].

4.2.1 English vs Arabic labels

The first experiment shows nearly identical training and validation losses for both multilingual models (see Figure 4.1a for GigaBERT and Figure 4.2a for mBert) with slightly better evaluation scores for models trained on Arabic labels. mBert shows a significantly lower loss compared to GigaBERT, but lower evaluation scores with COCO Caption evaluation (see Figure 4.1b for GigaBERT and Figure 4.2b for mBert). This result suggests that the validation loss is more correlated to the model configuration, than to the evaluation scores. Table 4.1 shows the exact final evaluation scores for all models.



(a) Training and validation loss

(b) Evaluation scores

Figure 4.1: (a) Training and validation losses for GigaBERT trained on English vs Arabic labels. (b) Respective evaluation scores over all epochs. Scores for models trained on English label is marked with a solid line and Arabic labeled scores are marked with dashed lines.

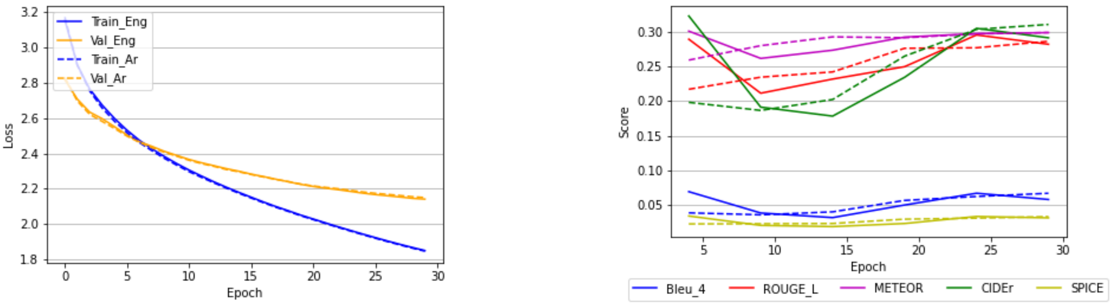
4.2.2 Learning Curve

All of the models, except AraBERT, show a strictly decreasing training and validation loss with increasing amounts of data (see Figures 4.3a, 4.4a and 4.5a for AraBERT, ArabicBERT and GigaBERT respectively). Note that GigaBERT trained on 100% of Flickr8k is identical to the model trained on Arabic labels in the previous experiment.

In the case of AraBERT, the 75% loss curves are way higher than the 100% and 50% curves, but the 100% loss curves are still lower than the 50% ones. The unstable training results of

[3] https://github.com/tylin/coco-caption

[4] https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/3



(a) Training and validation loss

(b) Evaluation scores

Figure 4.2: (a) Training and validation losses for mBert trained on English vs Arabic labels. (b) Respective evaluation scores over all epochs. Scores for models trained on English label is marked with a solid line and Arabic labeled scores are marked with dashed lines.

Model	Labels	BLEU-4	ROUGE-L	METEOR	CIDEr	SPICE
GigaBERT	English	**0.074**	**0.29**	0.3	**0.33**	**0.037**
	Arabic	0.062	**0.29**	**0.31**	0.31	**0.037**
mBert	English	0.058	0.28	0.30	0.29	0.031
	Arabic	0.067	**0.29**	0.30	0.31	0.033

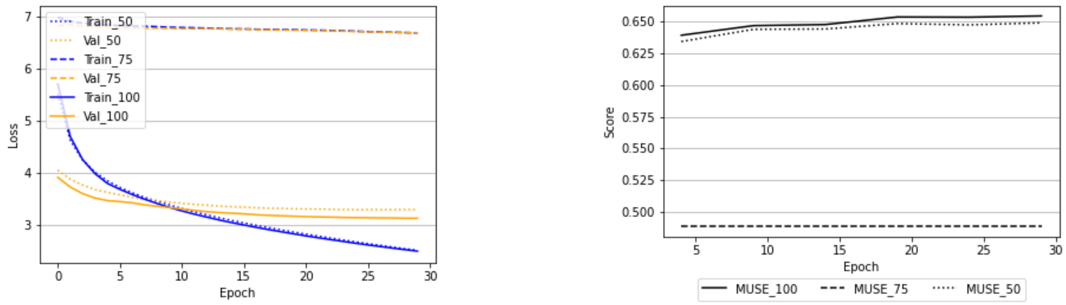
Table 4.1: Evaluation scores (evaluation on epoch 30) for the trained models. The best scoring models are marked in bold for each evaluation metric.

AraBERT suggest that the chosen learning rate is too large. The results from the learning rate grid search is shown in Figure 4.6, for AraBERT and GigaBERT. An additional experiment with AraBERT trained on 75% of the data with the smaller learning rate of $5e^{-5}$ shows a much more stable learning curve than the one shown in Figure 4.3a.

In addition, all of the models from the learning curve experiment were evaluated with MUSE to investigate the correlation between semantic scores and an increased amount of data. The evaluation over training time is shown in Figures 4.3b, 4.4b and 4.5b for AraBERT, ArabicBERT, and GigaBERT respectively. In general, more data increased evaluation scores. One notable thing is that the final score of GigaBERT trained on 75% of data outperformed 100%, but Figure 4.5b shows that the 100% curve is generally higher than the 75% curve. This finding suggests that the average MUSE score has a high variance. Table 4.2 shows the final MUSE scores for each model.

4.2.3 Large Scale Training

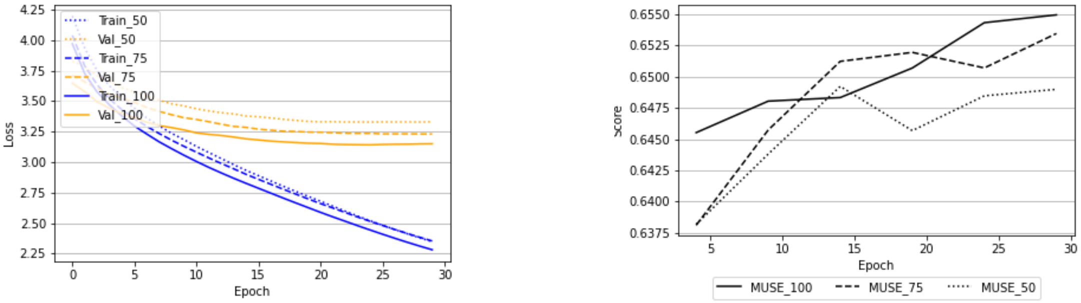
Table 4.3 presents the final test scores of a selection of our models, and models previously proposed by Al-muzaini et al. (2018), Afyouni et al. (2021) and ElJundi et al. (2020). Out of the previous works, only the model by ElJundi et al. (2020) is tested on the same Flickr8k test set as ours. All of our models are named after the scheme *modelBatchSize-dataset*, where



(a) Training and validation loss

(b) MUSE scores

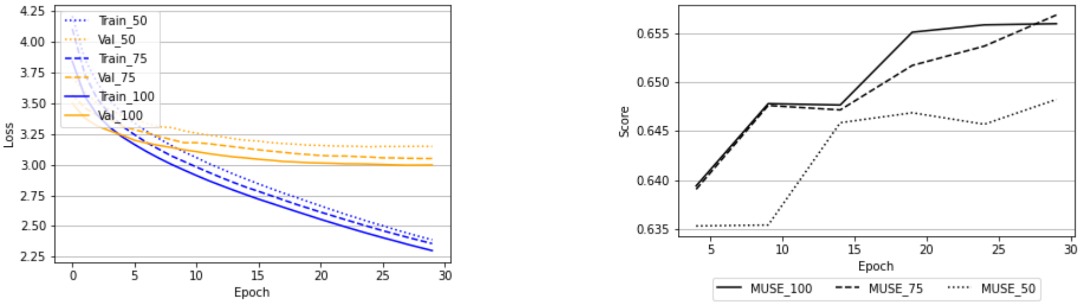
Figure 4.3: (a) Training and validation losses for AraBERT trained on 50%, 75% and 100% of Flickr8k. (b) MUSE evaluation scores over all epochs.



(a) Training and validation loss

(b) MUSE scores

Figure 4.4: (a) Training and validation losses for ArabicBert trained on 50%, 75% and 100% of Flickr8k. (b) MUSE evaluation scores over all epochs.

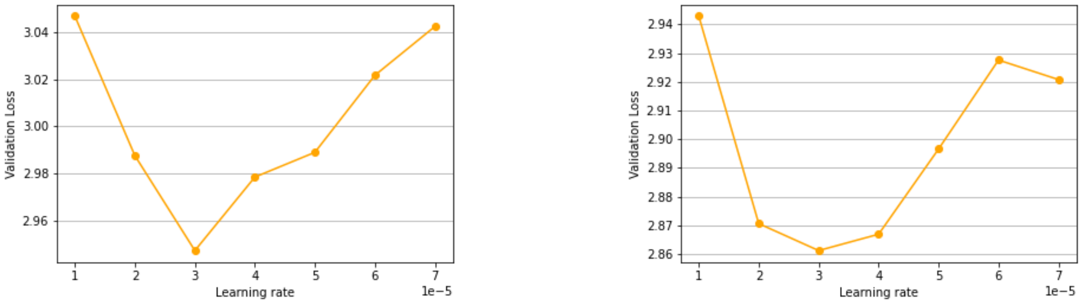


(a) Training and validation loss

(b) MUSE scores

Figure 4.5: (a) Training and validation losses for GigaBERT trained on 50%, 75% and 100% of Flickr8k. (b) MUSE evaluation scores over all epochs.

model is our initialization model, *BatchSize* is the training batch size and *dataset* is the dataset trained on. For example, one of our best performing models was initialized on AraBERT and



(a) AraBERT validation loss

(b) GigaBERT validation loss

Figure 4.6: Linear grid search optimization of learning rate η for two models (a) AraBERT and (b) GigaBERT. For both models, the optimum is found at $\eta = 3e^{-5}$. The grid search was in the interval $\eta \in [1e^{-5}, 7e^{-5}]$, and aimed to minimize model validation loss when trained on Flickr8k. All validation losses are from the 30:th and last epoch, i.e. there was no remarkable overfitting during training.

Percentage Model	50%	75%	100%
AraBERT	0.649	0.488	**0.655**
ArabicBERT	0.649	0.653	**0.655**
GigaBERT	0.648	**0.657**	0.656

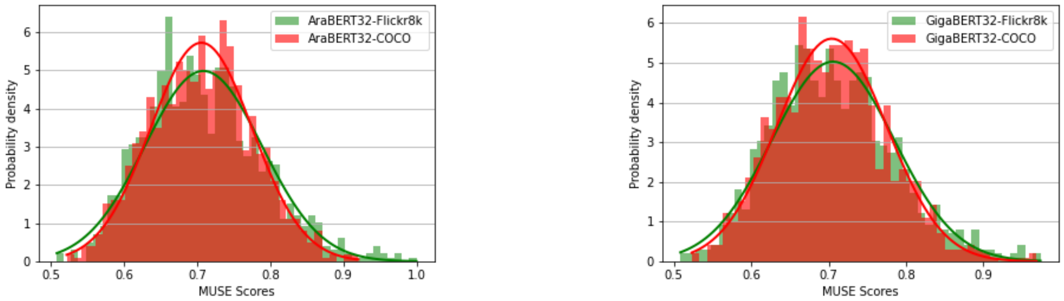
Table 4.2: Final MUSE scores for each model, evaluated on epoch 30. The highest scores are marked in bold.

trained with a batch size of 32 on Flickr8k. Therefore, we named the model AraBERT32-Flickr8k. AraBERT32-Flickr8k outperforms the model by ElJundi et al. (2020) on all BLEU scores, and most remarkably on BLEU-4, where we see a 61.4% increase. We chose to drop the SPICE scores from the table because of the evaluation scripts incompatibility with the Arabic language, as later discussed in Section 5.4.

Model	Test set	B1	B2	B3	B4	ROUGE-L	METEOR	CIDER	MUSE
Jindal (2018)	Flickr8k	0.658	0.559	0.404	0.223	-	0.201	-	-
Al-muzaini et al. (2018)	COCO & Flickr8k	0.462	0.260	0.190	0.080	-	-	-	-
Afyouni et al. (2021)	COCO	0.649	0.413	0.241	0.136	0.470	0.408	-	0.78
ElJundi et al. (2020)	Flickr8k	0.332	0.193	0.105	0.057	-	-	-	-
AraBERT32-Flickr8k		**0.391**	**0.246**	0.150	0.092	0.331	0.314	0.415	**0.671**
AraBERT32-COCO		0.365	0.221	0.129	0.0715	0.310	**0.317**	0.36	0.669
AraBERT256-Flickr8k	Flickr8k	0.387	0.244	**0.151**	**0.093**	**0.334**	0.312	**0.428**	0.668
GigaBERT32-Flickr8k		0.386	0.241	0.144	0.0827	0.331	0.315	0.403	0.669
GigaBERT32-COCO		0.36	0.215	0.124	0.0708	0.308	0.311	0.344	0.668
	Δ	**0.059 ↑**	**0.053 ↑**	**0.046 ↑**	**0.036 ↑**				

Table 4.3: Our model scores compared to previous models. The highest scores on our test-split are marked in bold. Of all the previous ones, only the model by ElJundi et al. (2020) uses the same test-split as us. Other test-splits are unknown.

To get an idea of how well our models capture object semantics, we plotted two histograms, each comparing the MUSE-distribution of two models. Figure 4.7a shows the final distributions of the test split inferred on AraBERT32-Flickr8k and AraBERT32-COCO, while Figure 4.7b shows the final distributions of the test split inferred on GigaBERT32-Flickr8k and GigaBERT32-COCO. Note that the mean values of the distributions are higher than the final score presented in Table 4.3, since we only counted the best performing captions for each image. Furthermore, note that the standard deviations of the models trained on Flickr8k are smaller than the models trained on COCO.



(a) MUSE score distribution for AraBERT

(b) MUSE score distribution for GigaBERT

Figure 4.7: (a) MUSE score distribution for the best performing captions of the test split inferred on AraBERT32-Flickr8k vs AraBERT32-COCO. (b) MUSE score distribution for the best performing captions of the test split inferred on GigaBERT32-Flickr8k vs GigaBERT32-COCO. Every bar represents the probability for a score being inside an interval of length 1%

We complement Table 4.3 with human evaluations according to the guidelines of THUMB (Kasai et al., 2021) described in Section 3.8. Figure 4.4 shows the top 6 best MUSE scoring captions of AraBERT32-COCO, with images and THUMB-scores. Figure 4.5 shows the bottom 6 MUSE scoring captions of AraBERT32-COCO. All of the evaluations were made by three experts in Arabic language.

In general, the human evaluations show accurate results. In the first row of Table 4.4, the candidate caption:

رجل يرك دراجة ترابية فوق تلة صخرية
"Man riding a dirt bike on a rocky hill"

is nearly perfect. It is almost identical to the reference caption:

رجل يركب دراجة ترابية فوق بعض الصخور
"Man riding a dirt bike over some rocks",

and only differs in the last phrase. The candidate caption on the second row

> كلب في الماء مع كرة في فمه
> "A dog in the water with a ball in its mouth"

also shows a nearly identical match to the reference caption

> كلب يخوض في الماء مع كرة في فمه
> "Dog wading in water with a ball in its mouth",

and was even preferred by the Arabic speaking experts.

Not all results were accurate. Looking at Table 4.5, the first row shows candidate caption

> مجموعة من الناس يتسلقون على ظهر شاحنة
> "Group of people climbing on the back of a truck",

while the closest reference caption مدينة ملاهي translates to "Amusement park". Though the candidate sentence is fluent and grammatically correct, it appears to be random in the context of the image. This shows how the models in these examples fails to identify objects in the image and correctly describe a scene. On a small note, the reference caption "Amusement park" is also very brief, and does not capture the whole image, compared to one of the other reference captions

> الناس في رحلة في مدينة ملاهي
> "People on a ride in a amusement park".

For complementary training statistics and the complete table with scores for all trained models, see Appendix B section.

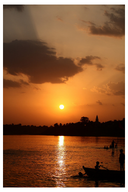
Image	Caption	P	R	Pen.	Total
	Candidate caption: (MUSE 0.920) رجل يركب دراجة ترابية فوق تلة صخرية Reference captions: رجل يركب دراجة ترابية فوق بعض الصخور رجل يرتدي تمويه يركب دراجة نارية رجل يلبس التمويه يوجه دراجة نارية عبر بعض الحجارة	5	4	0	4.5
	Candidate caption: (MUSE 0.911) كلب في الماء مع كرة في فمه Reference captions: كلب يخوض في الماء مع كرة في فمه الكلب البني يسبح في الماء مع كرة برتقالية في فمه الكلب البني في الماء ، مع الكرة البرتقالية والزرقاء في فمه	5	4	0	4.5
	Candidate caption: (MUSE 0.904) كلب ابيض صغير يركض عبر حقل مغطى بالعشب Reference captions: كلب أبيض صغير يجري في حقل عشبي كلب أبيض يمتد في الغابة مع آذانه ترفرف في النسيم كلب أبيض يركض على العشب ،	5	5	0	5
	Candidate caption: (MUSE 0.902) غروب الشمس فوق بحيرة عند غروب الشمس Reference captions: غروب الشمس فوق البحيرة هذا هو غروب الشمس الجميل على الماء مشهد مائي مع غروب الشمس في الخلفية	4	4	0.5	3.5
	Candidate caption: (MUSE 0.890) رجل يقف في الماء مع فيل Reference captions: رجل يقف فوق فيل في الماء رجل يقف على حيوان في الماء رجل يقف على فيل يرقد في بعض الماء	4	4	0	4
	Candidate caption: (MUSE 0.881) مجموعة من الناس وكلب يلعبون في المسبح Reference captions: الكلاب والناس يلعبون في بركة الناس والعديد من الكلاب في حمام سباحة في حديقة مجموعة من الكلاب والأطفال يلعبون في حمام السباحة	4	4	0	4

Table 4.4: Human evaluation of the top 6 MUSE scoring candidate captions of AraBERT32-COCO. Each candidate captions has three reference captions from the Flickr8k test-split. The reference captions with most similarity are marked first, and the other two are greyed out. THUMB-scores are shown to the right.

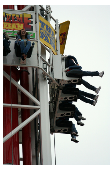
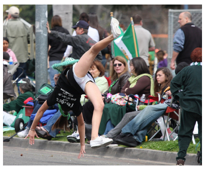
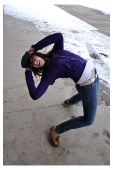
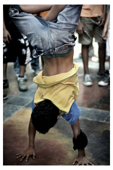
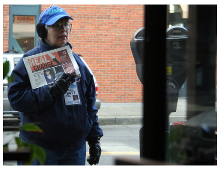
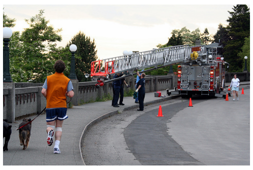
Image	Caption	P	R	Pen.	Total
	Candidate caption: (MUSE 0.490) مجموعة من الناس يتسلقون على ظهر شاحنة Reference captions: مدينة ملاهي الناس في رحلة في مدينة ملاهي شخص يضحك على متن السفينة الدوارة	3	3	0	3
	Candidate caption: (MUSE 0.495) شخص يقفز على لوح تزلج في الهواء Reference captions: يرتدي قميصا أسود حلة في الشارع بينما كان الحشد يراقبأجرى الشخص الذي كان فتاة تقوم برياضة في الشارع بينما يراقب الناس من الرصيف فتاة تقوم برياضة أمام حشد من الناس	2	2	0	2
	Candidate caption: (MUSE 0.501) امراة ترتدي قميصا ازرق وقبعة سوداء على الرصيف Reference captions: شخص ينحني للخلف فتاة تنحني للخلف حاملة قبعتها فتاة في بلوزة أرجوانية تنحني إلى الوراء	4	3	0	3.5
	Candidate caption: (MUSE 0.501) طفل صغير يرتدي سروال قصير وربطة عنق Reference captions: رجل يقف على يديه مع الكثير من الناس من حوله رجل يقف على يديه على رصيف بينما يراقب الآخرون صبي يقوم على الوقوف على اليدين في قميص أصفر	1	2	0	1.5
	Candidate caption: (MUSE 0.501) رجل يقف بجوار عداد وقوف السيارات Reference captions: امرأة تحمل عنوان صحيفة آتغيير حقيقيْ. امرأة تحمل جريدة تقول آتغيير حقيقيْ امرأة تحمل جريدة	2	3	0	2.5
	Candidate caption: (MUSE 0.502) رجل يرتدي قميصا برتقاليا يلعب لعبة مع كلبه Reference captions: امتدت سيارة إطفاء مزودة بسلالم على جانب الطريق تتوقف سيارة إطفاء وتجري مكالمة على جسر ، حيث يركض رجل وكلابه يقفز عداء ببطء مع كلب على شاحنة مزودة بسلم	4	3	0	3.5

Table 4.5: Human evaluation of the bottom 6 MUSE scoring candidate captions of AraBERT32-COCO. Each candidate captions has three reference captions from the Flickr8k test-split. The reference captions with most similarity are marked first, and the other two are greyed out. THUMB-scores are shown to the right.

Chapter 5

Discussion

In this chapter, we will discuss the results obtained and the methodology in the light of insights gained during this project. Additionally, we will discuss further experiments which could not be covered in detail in the given time frame of this project, and provide general comments about the subject. Finally, we will draw a conclusion.

5.1 Result findings

During our experiments, we evaluated several image captioning models trained on three different datasets: Flickr8k, Arabic-COCO, and a dataset combining these two. We initialized each model on an Arabic-specific BERT, before we trained them with the learning method OSCAR.

We report results better than the previous work on the Flickr8k dataset by ElJundi et al. (2020): 0.059, 0.053, 0.046 and 0.036 improvement on BLEU-1,2,3,4 respectively. Most surprisingly, the models trained on the smaller dataset performed better on the test set than the models trained on COCO, and even the combined dataset. All of the models trained on batch size 32 outperformed corresponding models trained on batch size 256 on every metric, with few exceptions.

5.1.1 English vs Arabic labels

Our first experiments show that both approaches, training on English and Arabic object labels, work in principle. Already at this stage, GigaBERT trained on English labels outperformed previous reported BLEU-1,2,3,4 scores with 0.0123, 0.0144, 0.0190, 0.0167 respectively. However, note that these scores were obtained from the val-split, and not the final test-split. We think that the reason to why GigaBERT with English labels outperforms Arabic labels is that the quality of the original English labels, in combination with GigaBERT's English pre-training, is much better than its machine translated counterpart. mBert is only trained on Wikipedia (Devlin et al., 2018), while GigaBERT is trained on the Gigaword corpus in addition to Wikipedia and web crawl data. This is how we explain GigaBERT's better performance. Moreover, the vocabulary of GigaBERT (21k English tokens vs 26k Arabic tokens) is richer and more balanced than the vocabulary of mBERT (53k English tokens vs 5k Arabic tokens), see Table 3.1.

Compared to GigaBERT, mBert showed a lower validation loss but lower evaluation scores, which suggests that validation loss is more correlated to the model configuration than to the evaluation scores. We still conclude that validation loss is negatively correlated to all of the evaluation scores, as expected, but that the correlation is not clear from the plots we made. We think this is the case because most of the learning is made during the first 5 epochs of training, while evaluation scripts are run every 5 epochs.

5.1.2 Learning Curve

Most importantly, this experiment proved that the model's object semantics aligning improved with the size of our dataset. But the experiment also gave us other insights, such as the importance of the learning rate η in BERT-based models. In the original BERT paper, Devlin et al. (2019) recommends learning rates of $5e^{-5}$, $3e^{-5}$ and $2e^{-5}$ for batch sizes of 16 and 32, when fine-tuning on NLP tasks. In a section of the OSCAR paper (Li et al., 2020), they initialized image caption fine-tuning on a BERT model (bert-base-uncased) directly without OSCAR pre-training, like we did in this project, to solve the nocaps task (nocaps: novel object captioning at scale) (Agrawal et al., 2019). In their case, they trained with a batch size of 256 and a learning rate of $3e^{-5}$. When we trained with a batch size 256 distributed on 8 GPU:s, we found the learning rate $9e^{-5}$ to decrease the final validation loss with 7% compared to the validation loss obtained when training with a learning rate $3e^{-5}$. We conclude that all the mentioned learning rates are valid, but we recommend a grid search optimization for best performance.

5.1.3 Large Scale Training

All of the presented results in this section outperform the scores previously reported by ElJundi et al. (2020). What is more interesting is that models trained on the smaller dataset (Flickr8k) beat models trained on the mixed dataset (COCO+Flickr8k), which in its turn beats models trained on the pure COCO dataset. This bias is probably caused by an object class imbalance between the datasets, which is discussed more in detail in section 5.5.2. In general, all of the models trained on batch size 32 outperformed the corresponding models trained on batch size 256 on every metric, but took longer to train. This result agrees with the observation that larger batches can cause a degradation in the quality of a model, as measured by its ability to generalize. One counter example is AraBERT256-Flickr8k, which marginally outperformed AraBERT32-Flickr8k on BLUE-3,4, ROUGE-L and CIDER.

All of our trained models showed similar scores in all categories. The AraBERT models trained on Arabic labels in general give slightly better scores than GigaBERT models trained on English labels. From this result, we conclude that a pure Arabic dataset (with Arabic labels) is to prefer, but do not exclude the possibility of pursuing bilingual captioning models in the future. As for the MUSE distributions shown in Figure 4.7, we see higher mean scores for models trained on Flickr8k, but also a slightly higher variance in comparison to models trained on COCO. We conclude that this observation is caused by the bias between the Flickr8k train- and test-split, as discussed in Section 5.5.2.

From the human evaluations, we see that MUSE with a cut-off is a good measure for how well the model aligns object semantics given an image: An average THUMB score of 4.25 with a MUSE cut-off greater than 0.881, see Table 4.4, and an average THUMB score of 2.67 with a MUSE cut-off less than 0.502, see Table 4.5. It would be interesting to plot the MUSE score of a candidate sentence against the recall of objects from that image, or the human evaluation. This way we could define more precise heuristic rules for "good" and "bad" MUSE scores.

For the large-scale experiments, we chose to not train ArabicBERT or mBert. Specifically, we deemed mBert to not produce good enough evaluation scores during training time. ArabicBERT showed similar MUSE scores to AraBERT and could probably produce similar evaluation scores for the final experiment, but because of time limitations and AraBERT's greater popularity, we chose to not continue with this model.

5.2 Lack of qualitative Arabic Data

OSCAR takes a large-scale pre-training approach. This approach differs from previous LSTM approaches, which can achieve significantly higher results than a BERT-based model for a small dataset on NLP tasks (Ezen-Can, 2020). The lack of qualitative data was a problem throughout the whole project, since there is not enough Arabic caption data publicly available relatively to the task.

To our knowledge, Arabic Flickr8k published by ElJundi et al. (2020) is the only human verified and publicly available Arabic caption dataset. Even though the dataset is human verified, captions on some images are questionable. For example, the image shown in Figure 2.2 contains an Arabic reference caption:

> رجل أسود على لوح ركوب الأمواج يركب موجة
> "Black man on a surfboard riding a wave",

which is very semantically different to the original English reference caption

> "A man in black on a surfboard riding a wave".

Furthermore, the publicly available Arabic-COCO used is purely machine translated and has to be verified by humans before employed in testing. The justification to why we still use machine-translated data is that we rely on the BERT-based language models to handle the grammar and syntax, while we count on the machine-translation model to correctly translate salient objects. The failure to do so leads to failing in learning image-text semantic alignments. For example, in our dataset, mistranslated object labels can be found. Some nouns are mistranslated into their homophone counterparts: "light" (*noun*) to "خفيفة" (*adjective*, bright; well-lighted), “block” (*noun*) to “منع” (*adjective*, to obstruct, or prevent someone or something) and so on. Li et al. (2020) shows that OSCAR learning curves for fine-tuning with object tags converge significantly faster than the methods without tags. In other words, high quality labels are crucial in image-text alignment for VL-pretrained models.

However, it is worth noting that to reach state-of-the-art English captioning results, it is not enough to translate and verify all of COCO captions and Flickr30k (559k+145k captions on 112k+29k images in total). The OSCAR+ model (Zhang et al., 2021) is trained on additional VL-datasets, such as 2.5M question-answer pairs for VQA.

5.3 Minimal pre-processing of data

During this project, we applied no pre- or post-processing of the Arabic raw text. This could have a negative effect on the performance of our models and the final results. In their Arabic image captioning work, ElJundi et al. (2020) writes:

> It is crucial to clean and pre-process our data before feeding it to any model because ‘garbage in, garbage out’...

They followed Arabic pre-processing techniques recommended by Shoukry and Rafea (2012):

1. Remove (harakat) diacritics.
2. Normalize the *hamza* (ء) on characters (for example to distinguish between a glottal stop and a mere vowel, hamza is usually added to letter *Alif* (ا) diacritically, either above (أ) or below (إ)).
3. Normalize some word ending characters, such as *taa marbouta* (ة) and *ya' maqsoura* (ي).
4. Remove punctuation as well as non-Arabic letters.

It is hard to say how this text processing scheme applied on our work would affect the final scores, but we think that a pre-processing scheme similar to the one above could give our models better performance. From the context of pre-processing point 2., our candidate caption output already seems to be hamza-normalized (i.e. all أ or إ → ا), while reference captions still contains extra hamzas on them. During evaluation, this of course affects the mean MUSE scores negatively, since the similarity function between symbols أ, إ or ا produces MUSE scores less than 1.

Another kind of Arabic text processing is *sub-word units segmentation* used in training some of the AraBERT models released by Antoun et al. (2020). The authors reduce the model vocabulary by segmenting words into into stems, prefixes and suffixes. For instance, “اللغة - *Alloga*” becomes “ال +لغ +ة - *Al+ log+ a*”. Since we chose bert-base-arabertv02, which is trained on non-segmented text, we did not use subword segmentation. Nonetheless, it would be interesting to see how segmentation applied to the candidate and reference captions would affect the evaluation scores.

5.4 Improved evaluation scripts

As stated before, we used the COCO Caption evaluation API for calculating most of the scores (BLEU-1,2,3,4, ROUGE-L, METEOR, CIDEr and SPICE). Out of all these metrics, SPICE does not seem to increase enough during training. On COCO with 5 reference captions, SPICE scores are generally in the range of 0.15-0.20, while our SPICE scores are in the 0.03-0.05 range.

We conclude that the out-of-the-box SPICE evaluation script is not compatible with the Arabic input. A way to fix this problem is to replace the Stanford Scene Graph Parser with an Arabic parsing pipeline, as mentioned in Section 3.8. The SPICE module uses Stanford CoreNLP for dependency parsing, which to this day does not support Arabic dependency parsing. However, a newer software created by the Stanford NLP Group, called Stanza, does support Arabic dependency parsing and could potentially be used to calculate SPICE scores. After these findings, we still chose to include SPICE in this report, since it is a well-established measure for image captions. With invalid SPICE scores, MUSE is the only metric that is purely semantics based.

The BLEU-1,2,3,4, ROUGE-L, METEOR and CIDEr scores are n-gram based and should work fine with Arabic input, though still penalized by the morphological complexity of the Arabic language, as discussed in Section 3.8.1. However, one question that remains is how n-gram mapping between candidate and reference captions are affected by the concatenative system of the Arabic language. It would be worth exploring how subword segmentation pre-processing, as described in the previous section, could improve the correlation between evaluation scores and human judgments.

We conclude that the state of caption evaluation tools today works fine with the Arabic language, but that we need more Arabic-specific tools to improve correlation between evaluation scores and human judgments.

5.5 Improved training

In the light of what has been discussed above, we propose further improvements we wish to apply, but that are outside the time frame of this project. We begin with proposing some direct improvements to our model, and then discuss the class imbalance between the datasets and how to counter it. Finally, we propose another angle to the Arabic image captioning problem worth exploring.

5.5.1 Direct improvements

Some direct improvement could be to initialize our models on a larger model with more parameters, for example bert-large-arabertv02. Another initialization worth trying out is XLM-RoBERTa, a pre-trained multilingual language model that is shown to significantly outperform multilingual BERT (mBERT) on a variety of cross-lingual benchmarks (Conneau et al., 2019). We made a quick initialization with XLM-RoBERTa (xlm-roberta-base) and showed that the model is not compatible with OSCAR out-of-the-box. It would be worth exploring possibilities to make XLM-RoBERTa OSCAR-compatible in the future.

A more extensive experimentation of the hyperparameters could also be made. Instead of only fine-tuning the initial learning rate η, we could in the future explore different learning rate schedules, with warm-up steps and different learning weight decays. We made grid search optimization for the learning rate on batch sizes 32 and 256, but with more powerful hardware, future experiments could try different learning rates on even higher batch sizes. Theory tells us that larger batch sizes allows computational speedups from the parallelism of GPUs. However, it is well known that too large batch sizes will lead to poor generalization. In our experiments, we obtained better scores for training on smaller batch sizes (32), with few exceptions.

One part of the captioning fine-tuning described in the OSCAR paper Li et al. (2020) that we did not explore was self critical sequence training (SCST) (Rennie et al., 2017). By applying CIDEr optimization (see Section 3.6.3) to our trained captioning models, we could increase evaluation scores even higher.

5.5.2 Dataset imbalance

Our experiments highlight another problem with our datasets: The data imbalance between Arabic-COCO and Flickr8k. These datasets are unbalanced in terms of data size, object vocabulary, and the number of annotations of each image. Arabic-COCO contains 89k images and 432k captions, while Flickr8k contains 6k images and 18k captions. The consequence of this size imbalance is that our model, when trained on both datasets, will have a bias towards the bigger one. This imbalance can be combated by sampling the smaller dataset with a sampling factor $N > 1$ to compensate.

The Arabic-COCO and Flickr8k vocabularies to label objects have quite different sizes and it shows in the final scores. By counting the unique object classes detected during image feature extraction, we found 495 unique classes in Arabic-COCO, 1101 unique classes in the train-split of Flickr8k, and an intersection of 242 classes between these two. In other words, although significantly smaller than its COCO counter part, Flickr8k contains a much richer and diverse set of objects. If we do not take this diversity into consideration, object alignments during training will not be sufficient for producing valid captions on the validation and test sets. Since both the validation and test sets are sampled from the same Flickr8k distribution as the train set, we expect a model trained on the train set to outperform a model trained on a poorer object vocabulary. We see this in the results, with models trained on Flickr8k outperforming models trained on bigger datasets. Table 5.1 shows the intersection between the sets of unique classes for all datasets used. Here we see that Flickr8k train-split cover 97.5% of Flickr8k test-split objects, while Arabic-COCO only covers 22.5%.

Object classes	Flickr8k Validation (816)	Flickr8k Test (788)
Flickr8k Train (1101)	792 (97.1%)	768 (97.5%)
Arabic-COCO (495)	179 (21.9%)	177 (22.5%)
Arabic-COCO ∪ Flickr8k Train (1354)	794 (97.3%)	773 (98.1%)

Table 5.1: Cardinality of intersections between different training datasets vs validation and test datasets. The table also shows the coverage percentage for each validation and test set. Note that the last row shows a slight improvement in coverage, but does not account for the dataset size imbalance between Arabic-COCO and Flickr8k. Also note that combining Arabic-COCO and Flickr8k only contributes with two more classes compared to only using Flickr8k on the validation set. In this case the two object classes were "Coffee table" and "Kettle".

5.5.3 Pre-training Arabic OSCAR

Our experiments with GigaBERT show that training Arabic captions on English labels is a working approach, which enables future language agnostic OSCAR models. Due to the lack of qualitative Arabic data, we suggest to pre-train a bilingual model, for example GigaBERT, on the same corpus as OSCAR+ (Zhang et al., 2021) (5.65M Images, 2.5M QAs, 4.68M captions and 1.67M pseudo-captions). The goal with this pre-trained model is to create a shared semantic space for object features and English object labels. By fine-tuning this pre-trained model on Arabic captioning data, we then bridge detected English labels into Arabic captions. Finally, the model should be fine-tuned using CIDEr optimization according to the recipe of Li et al. (2020).

5.6 Future work

Aside from the further work described in the previous section, we hope to see many contributions to the field of Arabic image captioning in the closest future.

As addressed in previous sections, machine translated Arabic labels should be verified by humans before further training on the datasets. This task should not be too expensive since it is 1,594 labels from the visual genome label map in total, but could greatly improve training. Secondly, the lack of qualitative Arabic data should be solved by translation and verification of all COCO captions, and then making the resulting dataset publicly available. As a suggestion, one could follow a crowd sourcing procedure as described by Al-muzaini et al. (2018), which includes some of the instructions that were used in the creation of COCO captions, and additional instructions specific to the Arabic language. This would create a new benchmark Arabic captioning dataset that we could train and test our models on.

Furthermore, we hope to see improved evaluation scripts. As discussed in Section 5.4, we need more tools for measuring the semantic correlation between candidate and reference captions. The first step would be to implement an Arabic dependency parser compatible with SPICE, see Section 5.4. Also, it would be interesting to plot the MUSE score of a candidate sentence against the recall of objects from that image or the human evaluation score. This way, we can define heuristic rules for MUSE evaluation in future experiments.

5.7 General comments

This section is dedicated to topics we think did not get enough space in this report, but still deserves to be mentioned and is relevant to the subject.

5.7.1 Machine translation vs end-to-end captioning

One might argue that machine translation is good enough to be directly applied to English state-of-the-art captioning, and that our end-to-end approach is redundant. The idea of an end-to-end captioning model is to generate more natural, native, results and eliminate the sources of error that might accumulate with machine translation. Machine translated captions are not reliable because of the many contextual errors Google Translate performs. For example, few machine translated captions contains a word that was translated literally and out of context, which makes the entire Arabic sentence incoherent.

The results of Jindal (2017) show that generating Arabic captions directly in one stage, produced superior results to a two stage English caption+Arabic translation process. To confirm that this is the case for our models, we should:

1. Fine-tune a OSCAR+ model on the English Flickr8k dataset.

2. Generate candidate test captions on our trained model.

3. Machine translate English candidate captions to Arabic.

4. Compare evaluation of translated captions with our directly generated Arabic captions.

ElJundi et al. (2020) did a similar experiment in their article, where they indeed confirmed that directly generated captions outperformed all BLEU scores of translated captions. These results confirm that an end-to-end approach is superior to the current state of machine translation, and that further development of end-to-end Arabic captioning is worth pursuing.

5.7.2 Language Agnosticism

Beside the Arabic datasets and the Arabic specific BERT used for initialization, all of the techniques in this report can be used for any human language. Our experiments confirm that OSCAR as described by Li et al. (2020) can be applied on other languages than English, and that a cross-lingual approach shows a lot of potential for future pre-trained models.

5.8 Conclusion

This work focused on Arabic image captioning using pre-trained bidirectional transformers. With this study, many conclusions can be drawn.

Firstly, we presented a method to adapt English state-of-the-art captioning models to other languages through public dataset benchmarks. Furthermore, we achieved results better than the previous work on the Flickr8k dataset by ElJundi et al. (2020). We also proposed working configurations and heuristics for hyper parameters in future experimentation on our proposed models.

Throughout this project, we gained many problem-specific insights about Arabic image captioning. The most prominent of them is that beside the lack of well-annotated datasets, the ones that are publicly available are very imbalanced in terms of object vocabulary and quality.

We conclude that the state of caption evaluation tools today works fine with the Arabic language, but that we need more Arabic-specific tools to improve correlation between evaluation scores and human judgments. This is especially true when it comes to semantics correlation, where, as of today, we only have MUSE.

We showed that pre-processing is not necessary for good caption generation, but we hypothesize that a pre-processing scheme similar to the one described in Section 5.3 could give our models a better performance.

We hope that our work will be useful for future Arabic image captioning models, and hope to see many contributions to the field in the closest future.

References

Afyouni, I., Azhara, I., and Elnagar, A. (2021). AraCap: A hybrid deep learning architecture for Arabic Image Captioning. In *ACLing 2021: 5th International Conference on AI in Computational Linguistics*.

Agrawal, H., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., Lee, S., and Anderson, P. (2019). nocaps: novel object captioning at scale. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Al-muzaini, H. A., Al-yahya, T. N., and Benhidour, H. (2018). Automatic arabic image captioning using rnn-lstm-based language model and cnn. *International Journal of Advanced Computer Science and Applications*, 9(6).

Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2016). Spice: Semantic propositional image caption evaluation. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 382–398, Cham. Springer International Publishing.

Antoun, W., Baly, F., and Hajj, H. (2020). AraBERT: Transformer-based Model for Arabic Language Understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference*.

Attai, A. and Elnagar, A. (2020). A survey on arabic image captioning systems using deep learning models. In *14th International Conference on Innovations in Information Technology (IIT)*, pages 114–119.

Banerjee, S. and Lavie, A. (2005). 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*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Multilingual bert readme. `https://github.com/google-research/bert/blob/master/multilingual.md`. [Online; accessed 6 Feb. 2022].

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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)*.

ElJundi, O., Dhaybi, M., Mokadam, K., Hajj, H., and Asmar, D. (2020). Resources and end-to-end neural network models for arabic image captioning. In *15th International Conference on Computer Vision Theory and Applications*.

Ezen-Can, A. (2020). A Comparison of LSTM and BERT for Small Corpus. *ArXiv*, abs/2009.05451.

Gage, P. (1994). A new algorithm for data compression. *C Users J.*, 12(2):23–38.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hodosh, M., Young, P., and Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. In *24th International Joint Conference on Artificial Intelligence*.

Huang, L., Wang, W., Chen, J., and Wei, X.-Y. (2019). Attention on Attention for Image Captioning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*.

Jindal, V. (2017). A deep learning approach for arabic caption generation using roots-words. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*.

Jindal, V. (2018). Generating image captions in Arabic using root-word based recurrent neural networks and deep neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 144–151, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kasai, J., Sakaguchi, K., Dunagan, L., Morrison, J., Bras, R. L., Choi, Y., and Smith, N. A. (2021). Transparent human evaluation for image captioning. *CoRR*, abs/2111.08940.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D., Bernstein, M., and Li, F.-F. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Lan, W., Chen, Y., Xu, W., and Ritter, A. (2020). An empirical study of pre-trained transformers for arabic information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., and Gao, J. (2020). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision – ECCV 2020*.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.*

Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Miller, G. A. (1995). Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Conference: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2017). Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, Los Alamitos, CA, USA. IEEE Computer Society.

Sabri, S. M. (2021). Arabic Image Captioning using Deep Learning with Attention. Master's thesis, University of Georgia.

Safaya, A., Abdullatif, M., and Yuret, D. (2020). KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media. In *14th International Workshop on Semantic Evaluation (SemEval2020)*.

Schuster, M. and Nakajima, K. (2012). Japanese and korean voice search. In *In International Conference on Acoustics, Speech and Signal Processing, pages 5149–5152.*

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword unit. In *CoRR, abs/1508.07909.*

Shoukry, A. and Rafea, A. (2012). Preprocessing egyptian dialect tweets for sentiment mining. In *The Fourth Workshop on Computational Approaches to Arabic Script-based Languages*, page 47.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Conference on Neural Information Processing Systems (NIPS).*

Vedantam, R., Zitnick, C. L., and Parikh, D. (2014). Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2016). Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431.*

Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (PMLR).*

Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Hernandez Abrego, G., Yuan, S., Tar, C., Sung, Y.-h., Strope, B., and Kurzweil, R. (2020). Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

You, Q., Jin, H., Wang, Z., Fang, C., and Luo, J. (2016). Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition.*

Young, P., Lai, A., Hodosh, M., and Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *Transactions of the Association for Computational Linguistics.*

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., and Gao, J. (2021). Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5588.

Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J. J., and Gao, J. (2019). Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence.*

Appendices

Appendix A

Experiment Hyperparameters

English vs Arabic labels

All experiments were trained and validated with the Flickr8k train- respective val-split. Table A.1 shows the exact hyperparameters for the experiments.

Model	Train	Object labels	Learning rate	Batch size	#Epochs
GigaBERT	Flickr8k	eng/ar	1e-4	32	30
mBERT	Flickr8k	eng/ar	1e-4	32	30

Table A.1: Hyperparameters used for the English vs Arabic labels experiments.

Learning curve

All experiments were validated with the Flickr8k val-split and trained on Arabic labels. Table A.2 shows the exact hyperparameters for the experiments. Grid search optimization was made on AraBERT and GigaBERT in the interval $\eta \in [1e^{-5}, 7e^{-5}]$ and a step size of $1e^{-5}$.

Model	Train	% of dataset	Learning rate	Batch size	#Epochs
AraBERT	Flickr8k	50/75/100	1e-4	32	30
Arabic-BERT	Flickr8k	50/75/100	1e-4	32	30
GigaBERT	Flickr8k	50/75/100	1e-4	32	30

Table A.2: Hyperparameters and datasets used for the learning curve experiments.

Large scale

All experiments were validated and tested with the Flickr8k test- respective val-split, and trained on Arabic labels. Table A.3 shows the exact hyperparameters for the experiments.

Model	Train	Object labels	Learning rate	Batch size	#Epochs
AraBERT	Flickr8k	ar	3e-5	32	30
	Arabic-COCO	ar	5e-5	32	50
	Arabic-COCO+Flickr8k	ar	3e-5	32	50
	Flickr8k	ar	5e-5	256	30
	Arabic-COCO	ar	9e-5	256	50
	Arabic-COCO+Flickr8k	ar	9e-5	256	50
GigaBERT	Flickr8k	eng	3e-5	32	30
	Arabic-COCO	eng	3e-5	32	50
	Arabic-COCO+Flickr8k	eng	3e-5	32	50
	Flickr8k	eng	9e-5	265	30
	Arabic-COCO	eng	9e-5	265	50
	Arabic-COCO+Flickr8k	eng	9e-5	256	50

Table A.3: Hyperparameters and datasets used for the large scale experiments.

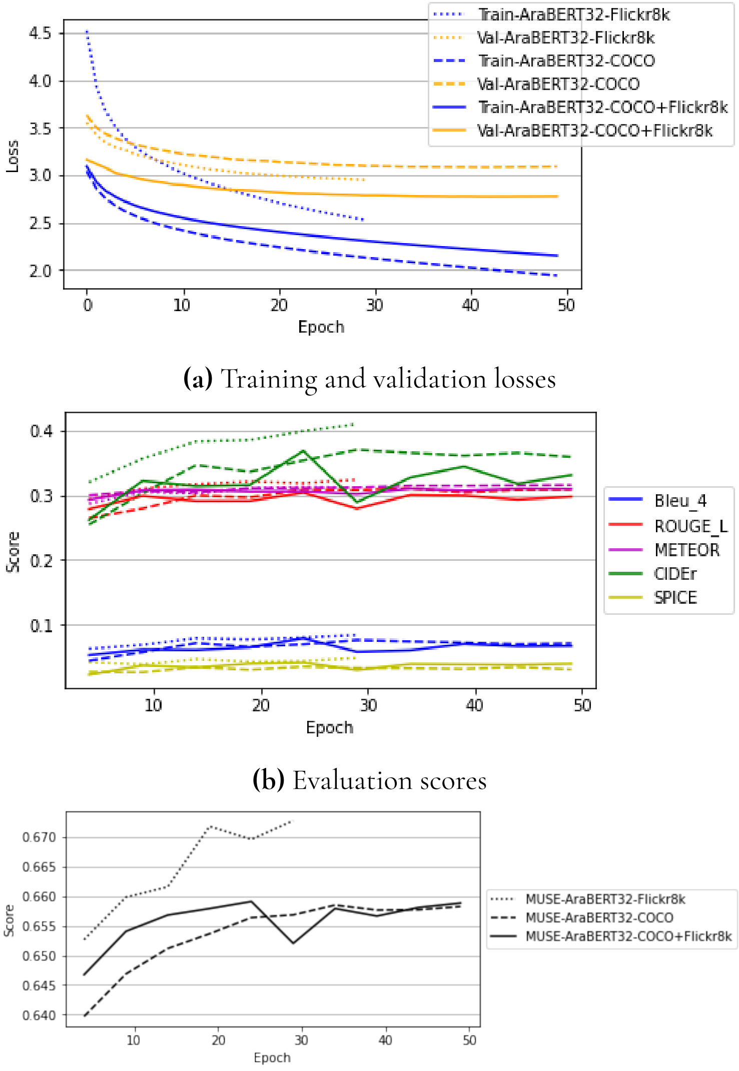
Appendix B
Complementary Results

Table B.1 shows scores for all models trained during the last experiment.

Model	Test set	B1	B2	B3	B4	ROUGE-L	METEOR	CIDER	MUSE
Jindal (2018)	Flickr8k	0.658	0.559	0.404	0.223	-	0.201	-	-
Al-muzaini et al. (2018)	COCO & Flickr8k	0.462	0.260	0.190	0.080	-	-	-	-
Afyouni et al. (2021)	COCO	0.649	0.413	0.241	0.136	0.470	0.408	-	0.78
ElJundi et al. (2020)	Flickr8k	0.332	0.193	0.105	0.057	-	-	-	-
AraBERT32-Flickr8k	Flickr8k	**0.391**	**0.246**	**0.150**	0.092	0.331	0.314	0.415	**0.671**
AraBERT32-COCO		0.365	0.221	0.129	0.0715	0.31	**0.317**	0.36	0.669
AraBERT32-COCO+Flickr8k		0.358	0.216	0.127	0.0715	0.317	0.316	0.364	0.661
AraBERT256-Flickr8k		0.387	0.244	0.151	**0.093**	**0.334**	0.312	**0.428**	0.668
AraBERT256-COCO		0.355	0.211	0.122	0.069	0.303	0.313	0.335	0.665
AraBERT256-COCO+Flickr8k		0.339	0.204	0.12	0.0686	0.302	0.31	0.339	0.655
GigaBERT32-Flickr8k		0.386	0.241	0.144	0.0827	0.331	0.315	0.403	0.669
GigaBERT32-COCO		0.36	0.215	0.124	0.0708	0.308	0.311	0.344	0.668
GigaBERT32-COCO+Flickr8k		0.362	0.216	0.127	0.0675	0.312	0.308	0.359	0.661
GigaBERT265-Flickr8k		0.376	0.235	0.141	0.0803	0.322	0.313	0.385	0.664
GigaBERT265-COCO		0.339	0.198	0.113	0.062	0.287	0.306	0.312	0.662
GigaBERT265-COCO+Flickr8k		0.365	0.217	0.128	0.0705	0.315	0.309	0.373	0.662

Table B.1: Our model scores compared to previous models. The highest scores on our test-split are marked in bold. Of all the previous ones, only the model by ElJundi et al. (2020) uses the same test-split as us. Other test splits are unknown.

Figure B.1 shows training statistics for the models AraBERT32-Flickr8k, -COCO and -COCO+Flickr8k. Figure B.2 shows training statistics for the models GigaBERT32-Flickr8k, -COCO and -COCO+Flickr8k.

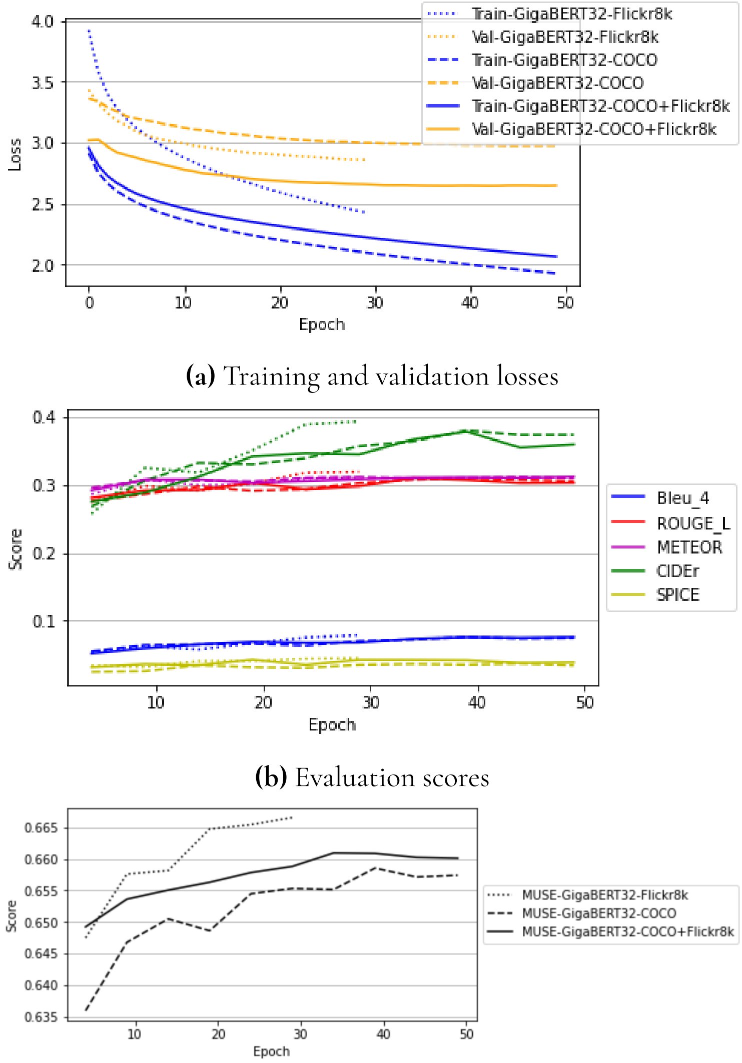


(a) Training and validation losses

(b) Evaluation scores

(c) MUSE scores

Figure B.1: (a) Training and validation losses for AraBERT32-Flickr8k, -COCO and -COCO+Flickr8k (b) Respective evaluation scores over all epochs. (c) Mean MUSE scores for all captions over all epochs.



(a) Training and validation losses

(b) Evaluation scores

(c) MUSE scores

Figure B.2: (a) Training and validation losses for GigaBERT32-Flickr8k, -COCO and -COCO+Flickr8k (b) Respective evaluation scores over all epochs. (c) Mean MUSE scores for all captions over all epochs.

INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTERAD 2022-02-15

EXAMENSARBETE Arabic Image Captioning using Pre-training of Deep Bidirectional Transformers
STUDENT Jonathan Emami
HANDLEDARE Pierre Nugues (LTH), Ashraf Elnagar (UOS), Imad Afyouni (UOS)
EXAMINATOR Jacek Malec (LTH)

Arabisk bildtextgenerering med hjälp av förtränande transformer-modeller

POPULÄRVETENSKAPLIG SAMMANFATTNING **Jonathan Emami**

Automatisk bildtextgenerering är idag ett utmanande problem inom datorseende och naturlig språkbehandling. Engelsk bildtextgenerering har sett stora framsteg de senaste åren, medan forskning på arabisk bildtextgenerering har hamnat efter. I detta examensarbete har vi utvecklat och utvärderat flera modeller för arabisk bildtextgenerering, alla initierade på förtränade transformer-modeller.

Bildtextgenerering har många olika tillämpningar, exempelvis effektiv bildsökning, auto-arkivering och som stöd för synskadade. De bästa bildtextgeneringsmodellerna idag följer en kodar-avkodar arkitektur:

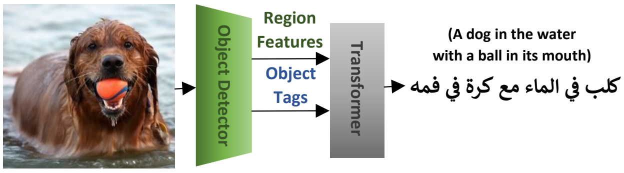
1. Extrahera den viktigaste informationen om bildens olika regioner m.h.a. en objektdetektor, t.ex. en CNN-kodare.

2. Generera en mening från den extraherade vektorn m.h.a. en språkmodell, t.ex. en RNN-avkodare.

I detta examensarbete använde vi förtränade transformer-modeller för att initialisera våra modeller för bildtextgenerering. Därefter finjusterade vi modellerna genom att träna dem på bild-text par med en inlärningsmetod som heter OSCAR. Denna inlärningsmetod använder sig av objekttaggar, detekterade i bilden, som ankarpunkt för att underlätta inlärningen av bild-text semantik.

Vårt examensarbete handlade om att utforska prestandan hos fyra olika transformer-modeller på ett bildtextdataset. De fyrade testade modellerna var *Multilingual BERT*, *AraBERT*, *ArabicBERT* och *GigaBERT*.

Våra resultat visar på bra inlärningsförmåga för alla våra modeller, men att AraBERT fick bättre evalueringspoäng. Figuren visar en bildtext genererad från AraBERT tränad på datasetet.


Object Detector
Region Features
Object Tags
Transformer
(A dog in the water with a ball in its mouth)
كلب في الماء مع كرة في فمه

Dessutom visade vi att det är möjligt att få bra resultat genom att träna flerspråkiga transformer-modeller, som GigaBERT, på arabisk bildtext med engelska objekttaggar. Däremot drar vi slutsatsen att en modell tränad på ett rent arabiskt dataset, med arabiska objekttaggar, presterar bättre.