Project report

Image Captioning for Automotive Domain

Federico Cocchi
289842@studenti.unimore.it

Paula Klinke
300997@studenti.unimore.it

February 23, 2022

Abstract

In past years the development of artificial intelligence models with a Transformer-like architecture took a big step forward in sequence modeling tasks. Only recently the scientific community starts to investigate the use of Transformer architectures also on multi-modal domains, like Image captioning. In this context, our work is located.

Our models were applied specifically to the automotive domain. To do so a dataset called '*cc_automotive*' was created, with about 155K triplets of data correlated with this domain.

The architecture used is a CLIP model to encode the relation of image regions. Taking those image regions a Memory - Transformer structure composed of an encoder and decoder to generate the final captions, related to the input image is used.

The Image captioning model can describe in natural language the concepts in the images, taking into account also the correlation among the objects and the semantic information.

Finally one can say that models trained on automotive related data perform better in the specific domain than models trained on a general dataset.

Here you can find the code of the project https://github.com/Paula-Kli/m3_new.

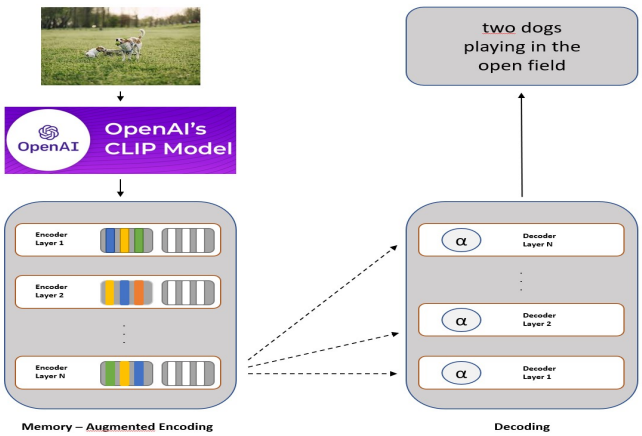

Figure 1: General architecture of our network

Contents

List of Figures

1 Introduction

An image captioning model can be applied to different scenarios. This type of model aims to describe the content of the image (visual information), using text sentences (natural language). Since this task is about two different types of data, it's a multi-modal scenario and which needs to be taken into account at implementation time. Image captioning systems can be divided into different sections. One part processes visual data, one works with natural language and another one merges the previous information from the two different domains.

To summarize at prediction time we start with an image as input and the final output is a generated caption that describes the input.

Image captioning can have different applications such as describing images to blind people, summing up the content of the image, or explaining what the system sees with its sensors to extend the explainability.

If we restrict the topic to the automotive domain [8], ADAS (Advanced Driving Assistance Systems) technologies are becoming increasingly important for vehicles and smart city applications. An approach which uses image captioning can probably better generalize some vertical problems such as traffic sign recognition or people detection to act on general traffic scenarios.

Using a unique multi-modal approach a traffic scene understanding can be provided and also the possibility to suggest behaviors of the environment can be investigated generating high-level semantic information for driving hints. Thus, in this work, we are aiming to get an impression of how well image captioning models work trained on automotive-related data in comparison to models trained on general datasets.

2 Related Work

The topic of image captioning can be tackled in many different ways as reported in the scientific literature [7].

With the advent of Deep Learning, image captioning tasks were provided by an RNN (Recurrent Neural Network) for the language model part and a CNN (Convolutional Neural Network) to encode the visual information [13], considering the CNN as a feature extractor. With this approach, we have 2 different situations at training and testing time. At training time, we have a *forcing teacher*, using the ground truth words until this point to predict the next one. Instead, at testing time we consider the word predicted at time t-1 to predict the one at time t, using different strategies (e.g. greedy, beam search).

A new state of the art was achieved combining the pre-train phase with Reinforcement Learning [17]. In this way, a caption is generated and a captioning metric is computed which is used as a reward. In this way, we can use a non-differentiable caption metric.

As we have seen before, in the beginning, CNNs were used to extract features from images but to improve the encoding and relationships among the objects in an image, image regions have been adopted extracted with an object detector [1] or with a more complex network as clip [15]. Clip has a huge knowledge of visual concepts and associate relations. These changes permit to have semantic and spatial information about the different parts of the image.

For a short time, the state of the art for image captioning tasks were represented by convolutional language models [2]. These kinds of models resolve one of the main downsides of the RNN. The implementation of RNN models has a sequential nature, while the convolutional ones manage parallel operations and by this increase the efficiency of the model.

Convolutional models were quickly overcome by Transformer. In our work we are using this type of architecture which represents nowadays SOTA.

A transformer is composed by an *encoder* and a *decoder*. The first one is composed of a stack of self-attention and feed-forward layers, the second one by self-attention with words and cross-attention using also the information from the encoder.

There are also some variations of this architecture; for instance, taking inspiration from BERT model [5], in order to create an *early-fusion* architecture. Encoding image and text at the same level, to generate the final caption connecting directly the two types of inputs [9].

Recently also a fully transformer architecture (CPTR) [12] was introduced. This architecture is based on an *encoder - decoder* structure but without using a feature extractor (ex. CNN) or an object detection.

The encoder takes the raw images and uses a ViT [6] whose outputs become part of the input for the decoder part. In the decoder part, the text, with the positional embedding, is taken into consideration too. The last feature of the decoder is considered to predict the final caption.

3 Our Dataset

In this work we investigated image captioning in the automotive domain. The benchmark usually used in literature for image captioning is the COCO dataset [11]. This dataset has 5 different captions associated with each image and in its last version contains 330 000 images. Figure 2 represents the distribution of the different categories present in COCO. As we can see there are quite a lot of images correlated with the automotive domain, but since we wanted to have a bigger dataset COCO did not provide enough data. From the best of our knowledge in literature there is no dataset for image captioning related to the automotive domain freely available. So we decided to create our dataset.

To do that we analyzed 3 different datasets for image captioning (YFCC100M, Conceptual Captions, WIT) and we decided to use the data from Conceptual Captions-3M [18]. We took these datasets to have, at the same time, a good number of images and a comparably good quality of captions. The instances were filtered to select only couples (image-text) related to the automotive domain. The Conceptual Captions dataset consists among other properties of images, captions and tags. Thus we filtered the tags according to the words: 'auto', 'car', 'automotive', 'street', 'road', 'parking', 'highway', 'semaphore', 'pedestrian', 'taxi', 'vehicle'. This created our dataset, which we called '*cc_automotive*' composed by around 155 000 couples of images and texts on the automotive domain.

To use with the webdataset library the dataset in the next phase, we created a .tar structure of all the data composed by contiguous repetition of triplets; composed of image, single caption and tags which in our case is empty since we do not use them for prediction.

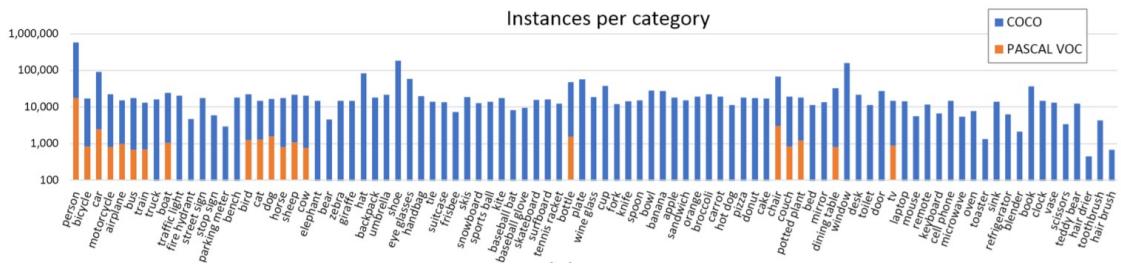

Figure 2: Class distribution in COCO dataset [11]

4 Our Implementation

For image captioning in this project, we used an enhanced version of the "Meshed-Memory-Transformer" [4].

The Meshed-Memory-Transformer has, different from a normal transformer implementation, slots to encode a priori knowledge too. The input of the Meshed-Memory-Transformer is preprocessed by an architecture that retrieves image regions.

In general, a Meshed-Memory-Transformer (m2) can be divided into an encoder and a decoder part. All connections are modeled as a scaled dot-product attention [19]. Therefore it acts on queries Q, keys K, and values V. First of all, the similarity is computed between queries and keys. The similarity gets scaled by the dimension d of the queries and keys. The result gets normalized and is used in a weighted sum with the value V vectors. Thus, attention is formally described by:

$$Attention(Q, K, V) = softmax\Big(\frac{Q \cdot K^T}{\sqrt{d}}\Big) \cdot V \tag{1}$$

When input image regions X are given with self-attention permutation invariant encodings of these image regions can be obtained like it is used in the transformer [19]. To receive queries, keys, and values the image regions are multiplied with the learnable weights W_q, W_k, and W_v. Using that as input for our attention we get a linear projection of the input vector X with the same cardinality as X. Formally written as:

$$S(X) = Attention(W_q \cdot X, W_k \cdot X, W_v \cdot X) \tag{2}$$

Since the self-attention only relies on pairwise similarity of input regions it can not model a priori knowledge. To overcome this problem the Meshed-Memory-Encoder introduces a memory-augmented attention operator. To model a priori knowledge, additional so-called "slots" are added to the keys and values. Thus, the key and value matrices get extended by a learnable matrix M each. By doing so $W_k \cdot X$ gets extended with M_k and $W_v \cdot X$ gets extended by M_v. This leaves the set of queries unaltered while the newly introduced slots which are independent of X can be learned.

If this is applied in a multi-head attention manner each head is learning independent projection matrices W_q, W_k, W_v and learnable slots M_k and M_v.

This memory-augmented operator then gets embedded into a Transformer-like layer. Doing so layers get stacked on top of each other. Thus, a-priori knowledge in combination with the possibility of building on features that are extracted in multiple levels from the before extracted image regions can be used.

The decoder of the Meshed-Memory-Transformer does not only has a cross-attention as it is expected in a transformer but rather it also has a meshed cross attention operator. The meshed attention operator can use the output of all encoder layers while computing the output sentence.

An input sentence Y gets connected with all outputs of the encoder layers $\tilde{X}$ which are then summed. Formally written as:

$$C(\tilde{X}^i, Y) = Attention(W_q \cdot Y, W_k \cdot \tilde{X}^i, W_v \cdot \tilde{X}^i) \tag{3}$$

This computed attention is modulated with a matrix that measures the absolute importance of a layer as well as the relative importance in comparison of this layer with respect to the others.

Decoding layers are applied in a multi-head fashion with a masked self-attention since the prediction of the next word should only depend on the last predicted words.

Until here the Meshed-Memory-Transformer has been described. As aforementioned our model is an enhancement of the previous model. From now on we are going to call the new version *m3*.

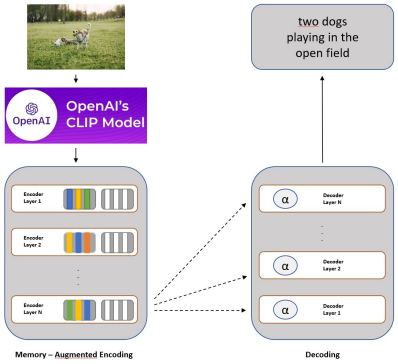

Figure 3: General architecture of our network

The model of m3 takes a lot of the characteristics of the Meshed-Memory-Transformer. Thus, it is a transformer-based architecture performing self-attention while having memory slots to be learned to be able to model a priori knowledge. Figure 3 shows that the main difference is that m3 does not have meshed attention in the decoder layers. This difference is justified by the small metrics improvement at the expense of a high computational power required for the mesh connections.

Additionally, a CLIP[15] (Contrastive Language–Image Pre-training) $RN50x16$ model is used as a feature extractor for the image regions which later are the input to the m3 model. CLIP provides different neural networks which are trained on a large number of images and their correlated texts found on the internet and the models can be freely downloaded and used under MIT License. While we use it only for the extraction of image regions the models of CLIP can be instructed in human language and have similar "zero-shot" capabilities as GPT-2[16] and GPT-3.

The input for our model is from CLIP extracted image regions and the correlated gt captions.

5 Training

In this section the metrics used and datasets used for validation are described. The metrics used for evaluation are CIDEr, BLEU, METEOR and ROUGE. The models get evaluated on a general COCO-validation dataset, a COCO dataset filtered for automotive and a subset of our cc_automotive dataset.

5.1 Training Metrics

The models were trained with a standard cross-entropy loss (XE). Therefore, the model needs to predict the next word given the already predicted ones.

The taken batch size for the average training was 25, but it was increased to 50 in the experiment with the big model. The training procedure was computed exactly in the different experiments, considering 70k steps of the trainer.

For prediction in the evaluation, beam search is used, with a beam size of 5. Using only the words with the highest possible score.

We evaluated all of our experiments with models using the following captioning metrics: CIDEr [20], BLEU [14], METEOR [3] and ROUGE [10].

Now we briefly describe these metrics:

- CIDEr (Consensus-based Image Description Evaluation): This metric aims to have a good correlation with the human judgment on the quality of the sentence. The implementation is based on the cosine similarity between Term Frequency & Inverse Document Frequency. The Gaussian penalty factor considers also the length similarity of the candidate and reference.

- BLEU: BLEU is designed for machine translation and is precision-oriented. The metric is based on n-grams and uses a weighted summation. Thereby we count whether certain words are mentioned in the machine-translated sentence which is also present in the ground truth sentence.

- METEOR: METEOR comes, like the previous metric, from machine translation and uses precision and recall, considering unigrams. In the F-mean recall is more important than precision.

- ROUGE: ROUGE is a recall-oriented metric designed for summarizing. This metric introduces the idea of an ideal overlapping between candidate and reference.

As a final output, metrics with a high correlation with the human judgment of the sentence are more taken into consideration.

5.2 Evaluation

For evaluation, 3 different types of data are taken into consideration and metrics for each experiment are computed using these validation sets.

- COCO-validation: This validation set is considered to be the standard. It is commonly used to compute benchmarks in Artificial Intellegence. This part of COCO used for validation uses COCO data in form of an image and a dictionary of 5 captions related to the image.

- COCO-automotive: Is a validation set taken from the COCO dataset. However, the data is filtered according to its labels. Therefore, we used the same automotive-related words we used for creating the cc_automotive dataset described in section 3.

- cc_automotive-validation: This is a split of data from our cc_automotive dataset. This validation set contains images that are related to automotive. Nevertheless, the quality of the captions is worse since cc_automotive is a subset from the Conceptual Captions 3M dataset [18] compared to the captions taken from the COCO dataset. Figure 4 shows a quantitative comparison of captions. This is just an example of how images taken from Conceptual Captions have worse related captions in comparison to captions on the COCO dataset.

5.3 Experiments

To test our proposed dataset we designed a list of experiments. These experiments are provided using cc_automotive compared to COCO as training data.

Figure 4: Comparison of ground truth captions provided by COCO versus Conceptual Captions

In the first one, a model is trained only with COCO data, to be also able to compare the results with other models in the literature.

In the second experiment, the previous checkpoints are used as a starting point, and based on that the model is fine-tuned using cc_automotive data. To do the model was trained on COCO data for 50 000 steps with a batch size of 25 and from there was trained for the remaining 20 000 steps on cc_automotive data.

In the third experiment, a model was trained from scratch using only cc_automotive data.

As mentioned before for each experiment data with captioning metrics was collected and the 3 different validation datasets were used.

As a last experiment, a bigger model was created (see default parameters in https://github.com/aimagelab/m3). In this experiment we tried to improve the result on coco-validation with respect to m3 COCO, to do this we have increased the batch size, the number of heads, and the dimension of the feed-forward layers.

Using different data during the experiments, but also different evaluation partitions a better comparison of the results taken can be acquired. For example in this way, the differences in the metrics using only a ground truth as in COCO-automotive and cc_automotive-validation can be evaluated better. Taking into consideration also the difference in the quality of the captions in the Conceptual Caption data versus COCO data, a different behavior based on the different quality of the ground truth captions is expected. As a last thing those different validation sets enable the possibility of comparing general images taken from COCO with images correlated with the automotive domain taken from COCO and cc_automotive.

6 Results

After having seen what validation sets are used and what metrics are acquired in this section we aim to not only present the results but also to discuss some of the findings.

6.1 Tables of Results

After having trained different models on different data now the scores those models reached trained on the different datasets are compared.

Figure 5: Comparison of different outputs from models trained on different data

First of all figure 5 shows a qualitative example taken from the trained models. It is one example of how models trained on different datasets perform. Here we can see that the model trained on automotive data captures the concept of emergency cars while the model trained on a general dataset like COCO describes the image correctly while not using the correct word "emergency". To see more examples have a look at figure 7.

Even though the metrics BLEU, METEOR, and ROUGE were collected and are presented in this report the focus in this work is on the CIDEr score of the different models.

Therefore we start to compare models based on the standard COCO-validation set with 5 captions related to each image.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	CIDEr
m2 [4]	81.6	66.4	51.8	39.7	29.4	59.2	129.3
m3 COCO	77.6	62.1	48.3	37.3	27.6	57.4	116.4
m3 finetuned	40.1	29.3	19.9	12.9	16.9	39.4	54.7
m3 cc_automotive	29.2	17.1	9.7	5.1	10.2	27.1	22.8
m3 big COCO	77.6	62.0	48.3	37.4	28.2	57.5	119.7

Table 1: Evaluation metrics considering coco-validation, benchmarks

In table 1 the results from the taken experiments are compared with the Meshed-Memory-Transformer. Considering as benchmark the split set with COCO-validation data. From a general point of view in this table, a similar trend on the COCO data between m2 and m3 (normal and big) models can be seen. M2 outperforms by 6% the other models trained on COCO data, but the training phase of m2 includes Reinforcement learning, which is not present for our models.

The diagram 8 shows that the CIDEr score on the COCO-validation set is the highest if the model is trained on COCO data too. Even if the model is mainly trained on COCO data and only fine-tuned on cc_automotive data it does not reach half of the CIDEr score of the first model. The model trained exclusively on cc_automotive data reaches the lowest score.

First of all it is evident that models trained primarily or fully on automotive data do not perform very well on a general dataset because it is out of domain. Even though this might be one explanation another one would be that the captions of the Conceptual Captions dataset

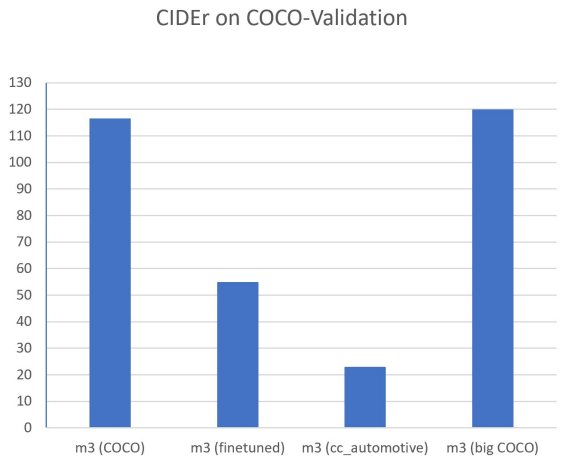

Figure 6: Comparison on COCO-based validation data using different training data
All m3 (except for "big") models are trained for 70 000 steps with a batch size of 25.

are not so good which results in worse predictions of models trained on that data compared to models trained on COCO data.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	CIDEr
m3 COCO	36.3	22.7	15.0	9.5	17.0	36.6	102.8
m3 finetuned	14.7	8.1	4.8	2.8	10.2	22.5	45.2
m3 cc_automotive	13.2	6.5	3.1	1.7	9.0	19.7	28.2
m3 big COCO	36.7	22.6	15.1	9.9	17.4	36.6	106.2

Table 2: Evaluation metrics considering coco-automotive, benchmarks

The table 2 shows results of the models evaluated on COCO data, which was filtered for the ones related to the automotive domain. Compared to the CIDEr data on COCO-validation, the model trained on cc_automotive performs better on COCO-automotive. This is caused by the training of the model which was on domain.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	CIDEr
m3 COCO	16.6	6.8	3.0	1.6	6.4	15.6	20.7
m3 finetuned	16.5	9.6	6.0	4.1	8.5	22.6	55.2
m3 cc_automotive	17.6	10.6	7.1	5.2	9.3	24.0	67.6
m3 big COCO	17.2	7.2	3.3	1.6	6.6	15.9	22.6

Table 3: Evaluation metrics on cc_automotive-validation, benchmarks

Table 3 shows that the model trained directly on the cc_automotive dataset has the best CIDEr score on the cc_automotive-validation set. Even the bigger model which was trained on COCO data scored a lot less on this validation set while the model that saw both datasets scored more than twice as good as both models trained only on COCO.

The validation sets of cc_automotive and COCO-automotive have only one ground truth caption whereas COCO-validation has five. To get evidence that it is reasonable to compare the models on the different validation sets a fourth validation set was taken into consideration. This was a subset of COCO which also has only one ground truth. The results of the CIDEr

scores of the different models are comparable to the ones of COCO-validation while the scores of BLEU, METEOR, and ROUGE are influenced by the number of captions associated with each image (see section 8 table 4). This underlines our choice of comparing the different CIDEr scores of the tables.

6.2 Discussion

After having had a short look at each validation set on the different experiments we want to combine the results. Therefore a few interesting examples are considered.

At first COCO-automotive versus cc_automotive-validation results are compared. When comparing the resulting CIDEr scores of the different models one can see that even though both of those metrics measure on automotive-related data the models trained on COCO in the first place also score better on images and captions taken from the COCO dataset. Both models trained only on COCO data even score worse in cc_automotive-validation compared to the model trained on cc_automotive and the CIDEr on COCO-automotive.

Another interesting result was comparing the small m3 version trained on COCO comparing the CIDEr score on COCO-validation versus the CIDEr score on COCO-automotive. While the CIDEr on COCO-validation is 116.4 the score on COCO-automotive is only 102.8. Drawing the conclusion that a model trained on general data performs very well on general data while it may not be so specialized on the domain and thus scoring less on a specific domain.

In the end, the model trained only on cc_automotive scores the worst on all validation sets except for the one taken from the dataset it was trained on, respectively cc_automotive-validation. Thus we can deduce that the model performs well on images with given captions with a similar structure and quality.

7 Conclusion

Having had an overview of all the experiments made and metrics compared now the results of the project are presented.

We can found three dimensions in which the experiments differ: the domain of the validation datasets, the number of captions associated with each image, and the difference in the quality of the ground truths.

The domain of the validation data plays an important role while it is not the most important one when looking into the CIDEr scores the different models achieved.

The second one is that the number of captions associated with each image changes the metrics except for CIDEr.

The most important dimension is the difference in the quality of ground truths used. Even though the model that was trained only on cc_automotive was already trained directly only on-domain it did not score a lot better on the COCO-automotive validation set with respect to the general COCO-validation.

To conclude, it is important to train a model on the domain it will be tested on later. Nevertheless, it is even more important to train it on similar data that you also expect to receive. While at the same time it was shown that better training data will give you better results at test time at least if you have around 160 thousand images as were extracted them from Conceptual Captions.

Video of the demo.

8 Additional Material

These data have the same characteristic of coco-validation a part of the number of elements inside the list of gt. If data in table 1 have 5 different captions for each image here in table 4 only a caption is provide for each image.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE	CIDEr
m3 COCO	38.6	24.7	16.4	11.3	17.6	38.4	115.2
m3 finetuned	17.8	10.1	5.8	3.3	10.6	25.8	51.8
m3 cc_automotive	11.4	5.4	2.6	1.3	6.1	16.5	21.9
m3 big COCO	39.0	25.0	16.7	11.6	17.9	38.5	118.5

Table 4: Evaluation metrics considering coco-test, benchmarks
It is a subset of COCO which provides only one caption per image

After showing some metrics on how the models perform, here we want to provide some additional qualitative examples of captions that were predicted by our model.

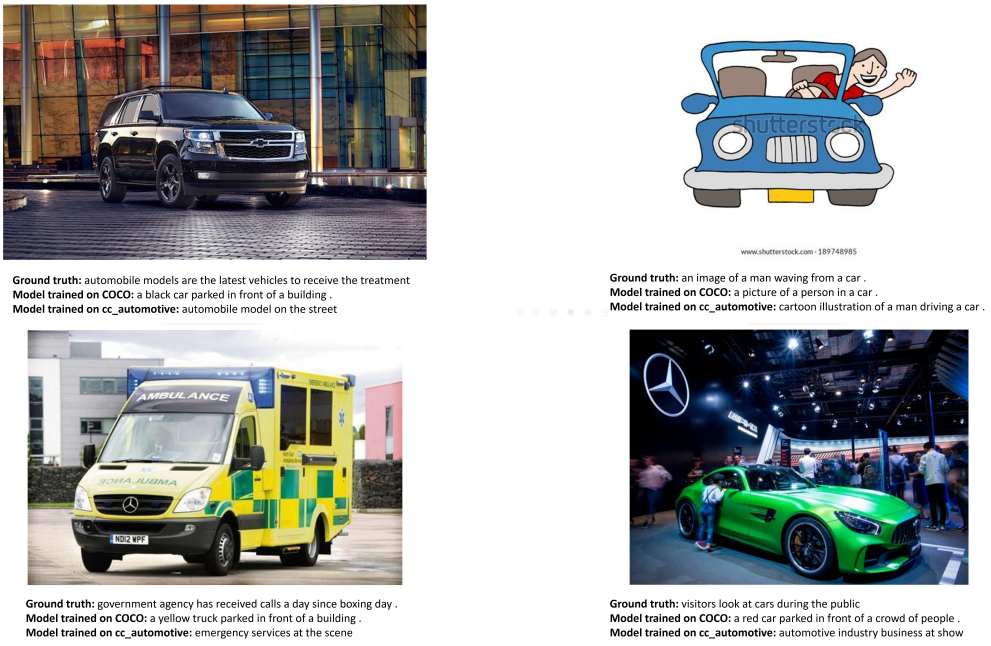

Figure 7: Comparison of predicted captions of the small models trained for 70 000 steps on COCO vs on cc_automotive data

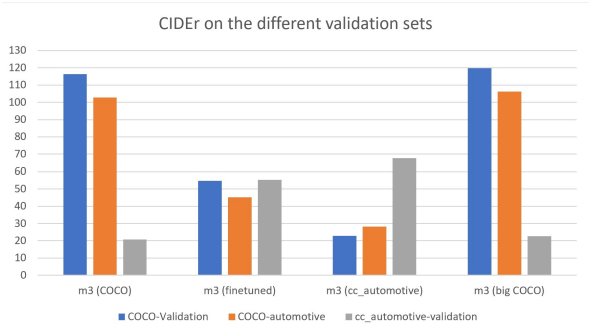

Figure 8: Comparison of models on all validation sets using different training data
All m3 (except for "big") models are trained for 70 000 steps with a batch size of 25.

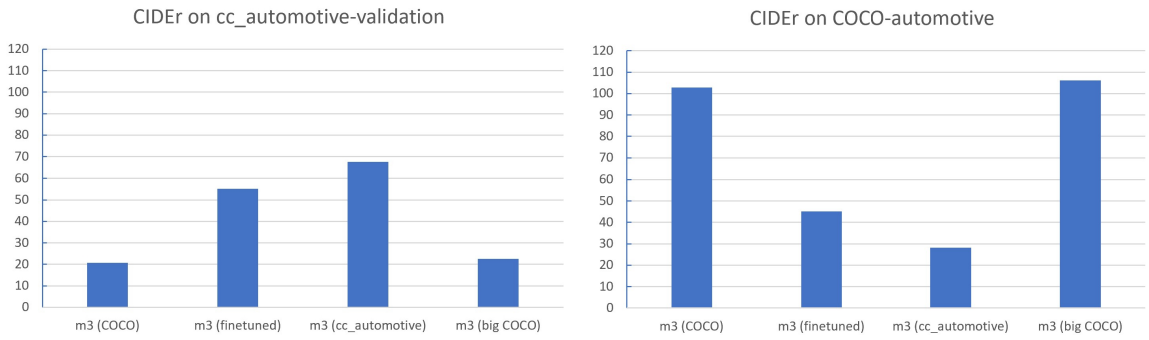

Figure 9: Comparison of different CIDEr values
All m3 (except for "big") models are trained for 70 000 steps with a batch size of 25.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[2] Jyoti Aneja, Aditya Deshpande, and Alexander G. Schwing. Convolutional image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] Satanjeev Banerjee and Alon Lavie. 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, June 2005. Association for Computational Linguistics.

[4] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-Memory Transformer for Image Captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[7] Md. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *CoRR*, abs/1810.04020, 2018.

[8] Wei Li, Zhaowei Qu, Haiyu Song, Pengjie Wang, and Bo Xue. The traffic scene understanding and prediction based on image captioning. *IEEE Access*, 9:1420–1427, 2021.

[9] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020.

[10] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[11] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[12] Wei Liu, Sihan Chen, Longteng Guo, Xinxin Zhu, and Jing Liu. CPTR: full transformer network for image captioning. *CoRR*, abs/2101.10804, 2021.

[13] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. Explain images with multimodal recurrent neural networks. *CoRR*, abs/1410.1090, 2014.

[14] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[16] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[17] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *CoRR*, abs/1612.00563, 2016.

[18] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[20] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726, 2014.