Milestone report Group 40

Willem Diepeveen
TU Delft
w.diepeveen@student.tudelft.nl

Daphne van Tetering
TU Delft
D.A.L.vanTetering@student.tudelft.nl

Niek van der Laan
TU Delft
N.vanderLaan-1@student.tudelft.nl

Paul Verkooijen
TU Delft
p.verkooijen@student.tudelft.nl

Kasper Wendel
TU Delft
k.wendel@student.tudelft.nl

1. Introduction

The canonical method for sequential modelling is the Recurrent Neural Network (RNN). In recent years, other architectures have been proposed and were shown to perform better in several cases. Especially using convolution or attention to model recurrence got more popular after the papers by Bai et al. [2] that proposes the Temporal Convolutional Network, and by Vaswani et al. [7] that proposes the Transformer network. The latter of the two networks is something special, as it does not use any recurrences nor convolutions making it very attractive due to its high degree of parallelism. The network is purely build up from attention mechanisms and fully connected layers. Although these networks have been tested against the accepted RNN variations, these transformer networks are not the standard yet.

2. Problem statement

In this work we want to test the Transformer network on yet another task: image caption generation. This problem entails generating a caption describing the picture given as input. Xu et al. [8] used a state of the art network consisting of a VGGNet and a LTSM to make an algorithm that turns an image into a caption in their paper *Show, Attend and Tell*. Previous research already showed that the Transformer can work very well on its own, but we want to know whether this still holds when it gets its input from the VGGNet. The central question will be whether the VGGNet-Transformer pair can outperform the network with the LSTM in terms of captioning performance. [1]

Our hypothesis is formulated as follows:

Does the Transformer network perform better on the image to caption task than the LSTM?

2.1. Dataset

During our research we will use the Flickr8k dataset. This dataset contains 8000 images with five captions each, where both the image and text data needs to be preprocessed differently. Each image-caption pair is used separately to prevent data from being unused. Presumably this also aids regularization since it prevents the network from taking only one of five captions into account. As text data, the provided lemmatized captions were used to simplify the vocabulary in the training data.

2.2. Expected Results

The goal of this project is to apply the Transformer network on an image to caption task. The short term goal of our group is to get the network to overfit on a small subset of the data to show the network is training and improving itself. The final goal will be to train the Transformer on the Flickr8k dataset and present these results on the 20th of June.

2.3. Evaluation

The evaluation will be done twofold: using the BLEU score and SPICE.

[1] Disclaimer: Zhu et al. [9] also investigated this topic. However, their implementation was not available and the paper was rather vague. Their ideas have been used as inspiration, but in order to have good comparison with *Show, Attend and Tell* we needed a somewhat different approach.

2.3.1 BLEU

BLEU scores [6] will be computed to enable performance comparison with the work of Vaswani et al. [7], which determines the similarity between source and target by matching n-grams.

2.3.2 SPICE

SPICE is a novel evaluation metric designed specifically for image captioning, presented in the work by Anderson et. al [1]. Unlike other metrics, like BLEU or CIDEr, SPICE determines the quality of a caption by analyzing its semantic content. To do so, all predicted and reference captions are formed into a scene graph that encodes objects, attributes and relationships found in each caption. The similarity between a generated caption c and its reference captions $S = s_1, ..., s_m$ is then defined as:

$$SPICE(c, S) = F_1(c, S) = \frac{2 \cdot P(c, S) \cdot R(c, S)}{P(c, S) + R(c, S)} \quad (1)$$

where P and R are Precision and Recall respectively. For further explanation on the SPICE-metric we recommend reading the work by Anderson et. al [1]. To compute the SPICE scores for our model we used the implementation provided by Anderson et al., which can be found here: `http://panderson.me/spice`.

3. Technical approach

Two distinct problems are required to be solved to answer the previously mentioned problem statement, namely preproccessing the Flickr8k dataset and building the proposed network. The approach and outcome to these problems are discussed below.

Firstly for each image, the VGGNet intermediate predictions are needed at the last convolution layer. These predictions are made in advance for each picture to reduce the computation times in the final image to caption network. This yields a new dataset where the image data is a matrix of 512×196. The following steps were taken to generate these predictions:

- Initialize a VGGNet with pretrained weights, reshape the last convolution layer output of size $14 \times 14\times$ to 512×196 and take this as the network output.
- For each image, resize it to 224×244 pixels as this is the required VGGNet input size.
- Generate the predictions for each image.

Secondly, the caption text needs to be preprocessed to a numerical format that can be used in the Transformer. We will discuss how this is done in more detail in the Intermediate Results section.

Finally, the dataset needs to be splitted into a training, testing and validation set. The dataset contains the original splits of [3], which were used to split the preprocessed data into distinct sets. The training dataset consists of 6000 images where each caption-image pair is used as a training object, yielding a total of 30.000 training objects. The same applies for the validation set of 1000 images, which thus consists of 5000 training objects. The captions for the testing data set are not splitted and thus this dataset contains 1000 images with each having 5 captions.

3.1. Network adaptions

The transformer implementation [2] is used as the decoder in our network. The encoder encodes the VGGNet intermediate predictions with a fully connected feedforward layer and a ReLU to a hidden state of some size d_{model}. This size will be determined by the word embedding. In particular, it corresponds to the maximum amount of words of all sentences in the dataset. Hence, d_{model} will be this maximum sentence length. Since we will switch the word embedding from SentencePiece to a pretrained as described in the intermediate results, this size has to be deterermined.

4. Intermediate Results

4.1. Preprocessing

The preprocessing stage consists of multiple steps to convert the caption text to a numerical format that can be used in the Transformer.

First, the lemmatized caption text (included in the dataset) have to be tokenized. Initially, an unsupervised text tokenizer for Neural Networks, SentencePiece [3], was used to simplify this process as this tokenizer learns the most used tokens/n-grams for a given vocabulary size. To incorporate this into our model, several steps had to be taken:

- Clean the caption text from any numeric characters, punctuation or single character words.
- Train the tokenizer on all available training captions for a given vocabulary size and encode the captions. This process also adds a start and end token to each caption.
- Find the longest caption in the training data and pad each caption to this length.
- Encode everything to one-hot vectors.

Since our method resulted in very large one-hot encoded vectors and was very complex, after consulting our TA, we decided to switch to using Word2Vec [4] [5]. This also deemed the use of SentencePiece unnecessary.

[2] https://github.com/Kyubyong/transformer

[3] https://github.com/google/sentencepiece

4.2. Final report

During the past weeks we kept our final report up to date to make sure all important decisions and results were included. This means that our final report already includes the Introduction, Theory and Method sections.

4.3. Next steps

In this section we will discuss the next steps taken to reach our goal. The first goal will be to show that our network has the ability to train itself (and overfit) on the dataset. To do so, we will change from using our own embeddings to Word2Vec and completing the model structure as a whole. This will be done by the end of the week, allowing us to retrieve the first results next Monday.

From that moment we will aim to improve our model by using techniques such as dropout and layer normalization to prevent the model from overfitting.

5. Reflection of topic/approach

During the past weeks we have gained a lot of knowledge regarding the inner workings of the Transformer and the implementation we are using. Even though our proposal seemed feasible, we have come to the conclusion that gaining a sufficient understanding of the Transformer has cost us a lot of time, more than anticipated beforehand. One thing that contributes to this is the fact that the original paper by Vaswani et al. is very high-level, requiring us to do a lot of additional research on parameter settings and input and output dimensions. Another fact contributing to this is our limited Tensorflow experience and knowledge. Even though Tensorflow was developed to easily create machine learning models, gaining an understanding of its inner-workings and its workflow cost us a lot some time.

Even though we have had some setbacks, we are still confident we will able to reach our goal.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[2] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[3] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

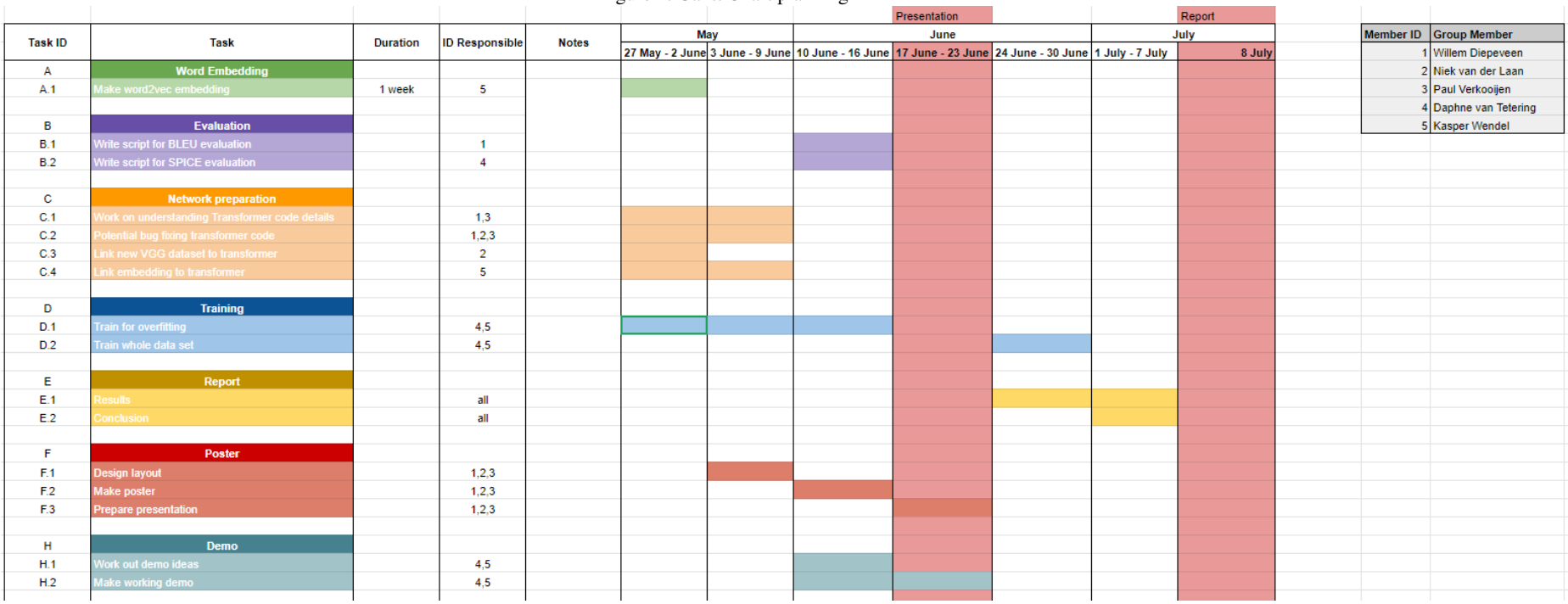
[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[8] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[9] Xinxin Zhu, Lixiang Li, Jing Liu, Haipeng Peng, and Xinxin Niu. Captioning transformer with stacked attention modules. *Applied Sciences*, 8(5):739, 2018.

Figure 1. Gantt Chart planning



Task ID	Task	Duration	ID Responsible	Notes
A	**Word Embedding**			
A.1	Make word2vec embedding	1 week	5	
B	**Evaluation**			
B.1	Write script for BLEU evaluation		1	
B.2	Write script for SPICE evaluation		4	
C	**Network preparation**			
C.1	Work on understanding Transformer code details		1,3	
C.2	Potential bug fixing transformer code		1,2,3	
C.3	Link new VGG dataset to transformer		2	
C.4	Link embedding to transformer		5	
D	**Training**			
D.1	Train for overfitting		4,5	
D.2	Train whole data set		4,5	
E	**Report**			
E.1	Results		all	
E.2	Conclusion		all	
F	**Poster**			
F.1	Design layout		1,2,3	
F.2	Make poster		1,2,3	
F.3	Prepare presentation		1,2,3	
H	**Demo**			
H.1	Work out demo ideas		4,5	
H.2	Make working demo		4,5	

Member ID	Group Member
1	Willem Diepeveen
2	Niek van der Laan
3	Paul Verkooijen
4	Daphne van Tetering
5	Kasper Wendel