Report for final project
Natural Language Processing 1

Dimitris Alivanistos
dimitris.alivanistos@student.uva.nl

Selene Baez Santamaria
selene.baezsantamaria@student.uva.nl

Sindi S——
sindi.s---@student.uva.nl

Charalampos Tamvakis
charalampos.tamvakis@student.uva.nl

Thanos ——-
thanos.----@student.uva.nl

Abstract

In this paper we create a system for image captioning. Using Show and tell as inspiration, we build an LSTM network. Processed image features as used instead of raw images.

1 Introduction

Two years ago, Google presented an end to end generative model capable to generate descriptive captions for any given image. This is an ongoing project, with new papers expanding on the model.

We try to replicate this model, within the best of our abilities.

2 Related work

Image captioning is a challenging task because it requires semantic understanding and correct expression of such understanding.

Previous work focused on interpreting images, and using language templates to describe them.

Show and tell is unique in that it approaches the problem from a machine translation perspective. In that sense, the task is to translate from visual language to Natural language, in this case English. The key to their success relies on data driven techniques and end-to-end training.

Their model consists on a 5 layers CNN followed by an Long Short Term Memory (LSTM) network. Included in the latter, are trainable word embeddings.

3 Our model

We focus on the second part of the task: expressing the semantic knowledge in a meaningful and grammatically correct manner. To do so, we use processed image features instead of raw images. We assume the features contain the semantic knowledge we need to convey, thus allowing us to explore in more detail the behaviour of NLP techniques.

We follow the main idea as Show and Tell, creating an LSTM network in order to obtain output word sequences of various lengths.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

A graph of the network is shown in Figure 1.

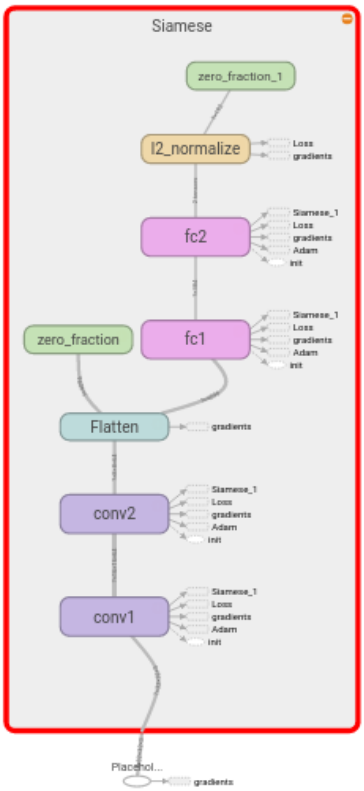


Figure 1: Network

3.1 Input

The image features were obtained by passing images through the VGG net, pretained on ImageNet. The features correspond to the fifth convolutional layer.

Attached to each image are five descriptive captions. Each caption consists of a single sentence. We processed this in order to create pairs of a single image and a single caption. All samples are seen by the network as independent from each other, thus enriching our datasets.

3.2 Vocabulary

Words from the training captions are counted and stored. With Zipf's law in mind, we assume that low frequency words in our sample are also rare in the English language. Therefore, and to control for an extremely large vocabulary size, words with less than n appearances in the total training set are discarded.

In the initial setup, we require a minimum of four appearances for a word to be kept in the vocabulary. Our reasoning is that if a word does not at least appear in all five captions of an image, then it is not essential for properly describing it. At the beginning, number of required appearances was chosen arbitrarily, however we experiment with different minimum and compare the effects it has on our results, but we analyze it further in section 5.

After counting and selecting the words to be included, we sort them by decreasing count, and store them for latter use.

3.3 Word encoding

Since the goal of this project is not to create optimal word embeddings, we encode words in the vocabulary in a basic way. On a first instance, a word is encoded as its index in the vocabulary. Later we use one hot encoding and compare results.

3.3.1 Start/Stop tags

We create special codes for representing the start and end of a caption. This serves as mechanism to signal the network to stop generating words.

4 Experimental setup

The network was created using Theano and Lasagne.

4.1 Data

The training set consists of 14,000 samples. The test set consists of 1,000 samples. Sample Images and their captions are shown in Figure 2

Figure 2: Sample image and its related descriptive caption.

We perform batch training.

4.2 Training

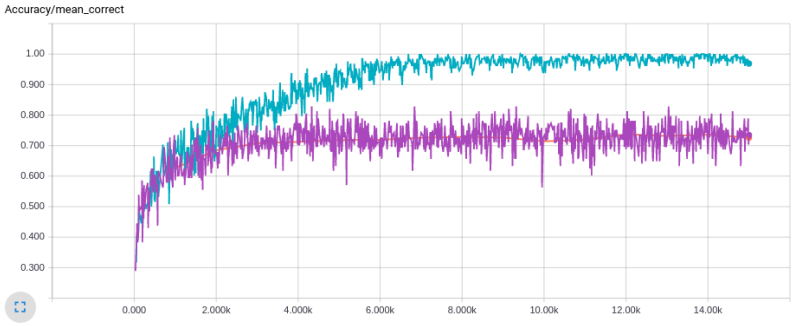
We train the implementation with the following parameters:

Parameter	Value
learning rate	2e-3
weight regularizer strength	0.
weight initlaization scale	1e-4
batch size	200
maximum steps	1500
dropout rate	0.
number of hidden neurons	'100'
type of weigh initialization	'normal'
type of weight regularizer	'l2'
activation function	'relu'
optimizer	'sgd'
loss	cross entropy

4.3 Evaluation

For evaluation we use the Baseline BLEU metric. According to Papineni et al [1], it provides a quick, inexpensive and language independent method of evaluating machine translation, so it provides a good mechanism for us to evaluate our captions. In our system we are using the Natural Language

Toolkit's (NLTK) implementation of the BLEU metric. Our captions are given a score between 0 and 1, where 1 means that the caption we generated was identical to our reference.

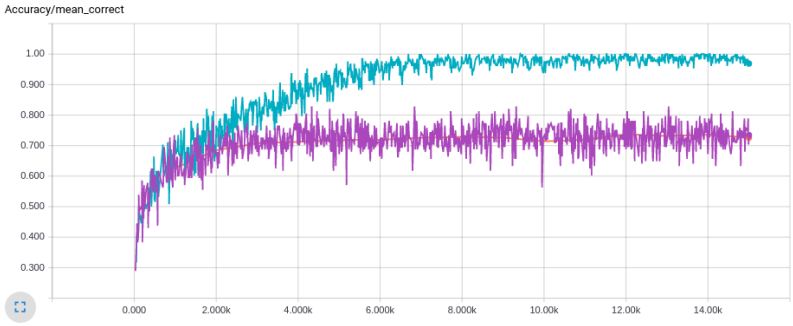


(a) BLUE scores here

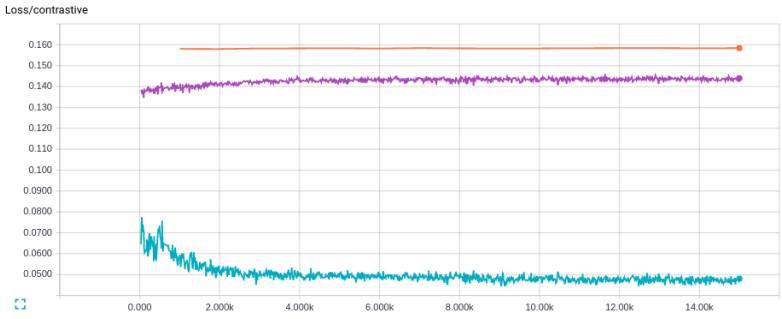
5 Results

5.1 Basic setup

The model performance is shown in Figure 4



(a) Accuracy



(b) Loss

Figure 4: Performance for basic setup. Blue line corresponds to training, purple line corresponds to testing.

5.2 Vocabulary size

We test different minimum count values for including words in the vocabulary. This modules both the length of and diversity in our vocabulary.

5.3 Word encoding

We compare index encoding against one hot encoding.

6 Conclusion

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow. org*.

References

[1] 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.