Image Caption Generation with Recursive Neural Networks

Christine Donnelly*
Department of Electrical Engineering
Stanford University
Palo Alto, CA
cdonnell@stanford.edu

1 Abstract

The ability to recognize image features and generate accurate, syntactically reasonable text descriptions is important for many tasks in computer vision. Auto-captioning could, for example, be used to provide descriptions of website content, or to generate frame-by-frame descriptions of video for the vision-impaired. In this project, a multimodal architecture for generating image captions is explored. The architecture combines image feature information from a convolutional neural network with a recurrent neural network language model, in order to produce sentence-length descriptions of images. An attention mechanism is used, which allows the network to focus on the most relevant image features at each time-step during caption generation.

2 Related Work

Most of the previous work in caption generation has involved the combination of a recurrent network (RNN) for language modeling and a convolutional network (CNN) for image feature extraction. In Ref. [1], a simple recurrent neural network was used for the language model, and the image context vector was only shown to the network at the initial timestep during caption generation. In Ref. [2], a recurrent neural network was also used, but the image context vector was shown to the network at each time-step. In Ref. [3], the authors used an approach similar to [1] but with an LSTM-based recurrent network. In Ref [4], the authors use an LSTM RNN and add an attention mechanism for spatially-distinguished image input at each time-step. In all of these works, the filter weights for the the convolutional neural network are taken from a pre-trained model, and are not altered during the training for caption generation. The authors of [3] found experimentally that attempting to train the CNN weights led to worse performance.

3 Technical Approach and Model

3.1 Baseline LSTM Model

In this work, the baseline model used to create image captions is a generative recurrent neural network in which the output word at time $(t-1)$ becomes the input word at timestep t. A Long Short Term Memory cell is used, with governing equations for the gates, cell state, and hidden state at timestep t according to:

$$f_t = \sigma(\mathbf{W}_f[x_t, h_{t-1}] + \mathbf{b}_f) \qquad \tilde{C}_t = tanh(\mathbf{W}_c[x_t, h_{t-1}] + \mathbf{b}_c)$$
$$o_t = \sigma(\mathbf{W}_o[x_t, h_{t-1}] + \mathbf{b}_o) \qquad C_t = i_t \odot \tilde{C}_t + f_t \odot C_{t-1}$$
$$o_t = \sigma(\mathbf{W}_i[x_t, h_{t-1}] + \mathbf{b}_i) \qquad h_t = o_t \odot tanh(C_t)$$

*Thank you to Andrei Iancu for providing access to GPU computing resources

where the input vectors x_t represent 200-dimensional word embeddings and the hidden state h_t is 500-dimensional. The output probability distribution over the full vocabulary is computed by projecting the hidden state h_t into the vocabulary space and computing a softmax:

$$y_t = (\mathbf{U}h_{t-1} + \mathbf{b}_u) \qquad\qquad p_t = \textbf{softmax}(y_t)$$

In order to incorporate image information, the 4096-dimensional image feature vector from the final fully connected layer of an Oxford VGGNet [5] (before projection into a 1000-dimensional space and softmax) is used as the initial input to the LSTM network. This feature vector is projected into the word embedding space with a trainable weight matrix: $x_{t=0} = \mathbf{W}_p w_{4096} + \mathbf{b}_p$. A schematic of the VGGNet convolutional network is shown in Fig. 1.

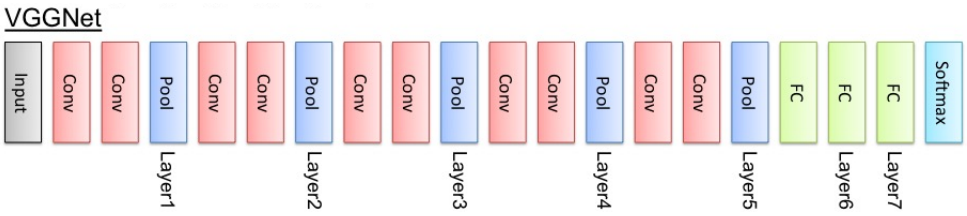


Figure 1: Architecture of the VGGNet CNN used in this work. For the baseline model without attention, image feature information is taken from Layer 7. For the attention model, image feature information is taken from the output of the convolutional layer before pooling in Layer 5. An alternative model was attempted with image feature information from the convolutional layer before pooling in Layer 2, but this yielded worse results.

3.2 LSTM Model With Attention

An alternative to the approach outlined above would be to introuce the image feature vector w at each timestep of the LSTM, and incorporate it into the update rules for the gates, cell state and hidden state:

$$\begin{aligned} f_t &= \sigma(\mathbf{W}_f[x_t, h_{t-1}, w_t] + \mathbf{b}_f) & \tilde{C}_t &= tanh(\mathbf{W}_c[x_t, h_{t-1}, w_t] + \mathbf{b}_c) \\ o_t &= \sigma(\mathbf{W}_o[x_t, h_{t-1}, w_t] + \mathbf{b}_o) & C_t &= i_t \odot \tilde{C}_t + f_t \odot C_{t-1} \\ o_t &= \sigma(\mathbf{W}_i[x_t, h_{t-1}, w_t] + \mathbf{b}_i) & h_t &= o_t \odot tanh(C_t) \end{aligned}$$

However, it has been found experimentally ([1], [3]), and was confirmed in this work, that performance is negatively affected if w_t at each timestep is the static 4096-dimensional fully connected layer feature vector. A potential alternative is to use an attention mechanism, whereby the network at each timestep outputs some signal indicating spatial feature information needed for the next timestep, and this information is used to compute the next image vector input w_t. The mechanism is depicted in Fig. 2.

The spatially distinguished feature information is taken from the last convolutional layer of an Oxford VGGNet. This layer consists of a 14x14 grid of 512-dimensional image feature vectors as depicted in Fig. 3, immediately after a ReLU nonlinearity and before pooling. Earlier layers of the VGGNet were also used in experiments, but yielded worse performance (it is likely that the lower-level edge and shape information captured in these layers is not useful when generating a brief summarizing sentence).

In Ref. [4], the attention mechanism proceeded by advancing the hidden state h_t through a multi-level feed forward network, which led to an output probability distribution over the 196 image locations. In this work, the hidden state h_t is fed through a single-layer feed-forward network consisting of a matrix multiplication and sigmoid nonlinearity to form a 512-dimensional feature mask vector:

$$mask_t = \sigma(\mathbf{W}_m h_t + \mathbf{b}_m)$$

The motivation is that the strongly activated indices of the feature mask should correspond to feature subtypes most relevant to the network at the next time-step. The inner product of $mask_t$ is taken

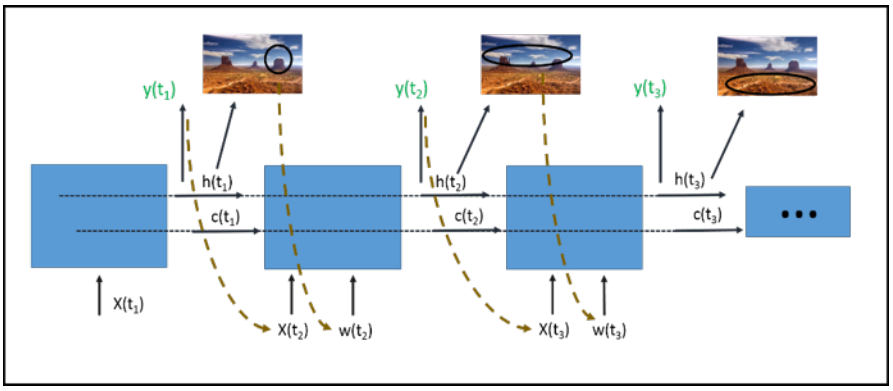


Figure 2: Visualization of LSTM network with attention, whereby the output word y_{t-1} becomes the input word x_t and the hidden state h_{t-1} is used to compute the input image vector w_t.

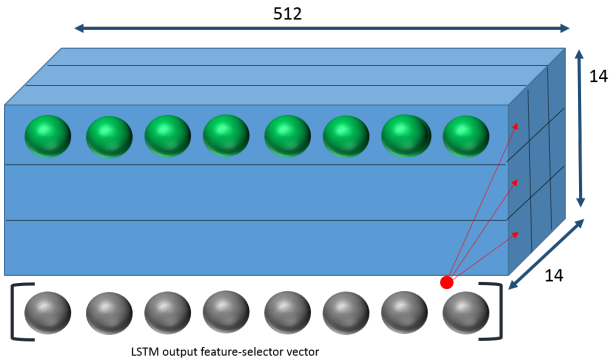


Figure 3: Depiction of the 14x14x512 CNN layer used in the attention model

with each of the 196 spatially distinuished feature vectors, and a softmax layer converts the results into a probability distribution:

$$p_v = \textbf{softmax}(mask_t \odot w_v)(v = 1...196)$$

The next image input then becomes:

$$\tilde{w_t} = \sum_{n=1}^{196} p_v \cdot w_v$$

Finally, it was found experimentally that it is beneficial (by an improvement of roughly 1.5 BLEU-1 points) to allow the $\tilde{w_t}$ to be projected into a different 512-dimensional space before input to the network, via a bias matrix and bias vector. Without this projection, the raw image vectors contain all positive values or zeros, whereas the other inputs at each timestep (x_t and h_{t-1}) contain both positive and negative values. Without the projection, it is likely that the σ nonlinearities become more easily saturated, which degrades the ability to train the network.

Thus, the final image input is:

$$w_t = (\mathbf{W}_{proj}\tilde{w_t} + \mathbf{b}_{proj})$$

In developing this model, a number of variations were tried, but discarded due to worse performance. These included:

1. A ReLU nonlinearity, or lack of a nonlinearity, for computing the image mask vector
2. A smaller network size (degraded performance was noted with hidden state size less than 400; little change was seen between 400 and 500).

3. A two-layer LSTM network: the addition of a second layer degraded BLEU-1 score by over 2 points

4 Experiment

4.1 Dataset

The MSCOCO dataset [6] was used for all experiments conducted in this work. This dataset consists of over 100,000 images, each with 5 corresponding captions which are generated by volunteers. The majority of captions consist of succinct descriptions with an average length of 10 words. Because of the volunteer nature of the project, language errors are relatively common (in a random sampling of 100 captions, five had either a spelling mistake or a significant grammar mistake).

Before training the models in this work, all caption characters were converted to lowercase, periods at the end of captions were removed (although commas and other punctuation occurring within the sentences were kept as separate tokens), and words occurring less than 3 times were removed from the vocabulary and replaced by the "unk" character. The resulting vobulary size was 13,679 unique words.

4.2 Details of Approach

During training, the training image captions were sorted by length into 16-caption minibatches of constant length (a larger batch size was desired, but program memory constraints associated with the large convolutional neural network outputs limited the batch size). At the beginning of each epoch, the captions composing the individual minibatches were randomly reshuffled. During one epoch of training, the network was shown the minibatches in a random order. The objective function was to minimize the average sequence cross-entropy error for a caption, given its corresponding image.

The stochastic gradient descent method was used for training, with adaptive learning rates for each variable according to the Adam algorithm. Dropout was applied at the input and output layers using the method suggested in [9]. The end of training was determined by evaluating the BLEU score for a 10,000-image subset of the MSCOCO validation split after each epoch.

During caption generation, the first network input at $t = 0$ was the projected image vector from the last fully-connected layer of the VGGNet. The second input was a special "begin" token, and at each subsequent timestep the output word at time t became the input at $t + 1$. Word generation continued until the "end" token was output by the network.

The word embedding size for the final model was 200-dimensional, with word embeddings kept trainable but initialized using GloVe vectors. The hidden state size was 500 dimensions.

In the final test run used to generate the captions and scores reported in the Results section, a Beam Search method with beam size of 8 was used to select the caption with the maximum average log-likelihood, as described in [3]. It was observed that the Beam Search improved the BLEU-2, BLEU-3, and BLEU-4 scores by about 2 points each, and the METEOR score by about 1 point, but had a negligible impact on the BLEU-1 score. This likely indicates that the Beam Search had a larger impact on improving the language accuracy (i.e. word ordering and grammar) than on selecting the correct object keywords corresponding to a given image.

4.3 Evaluation Metric

During training, the objective function is to minimize the cross-entropy error for an image/caption pair, but a different set of metrics are used for evaluation of final test results. As in machine translation, the BLEU score is the main metric used in previous works to evaluate image-to-text "translations". The BLEU-n score computes a modified n-gram precision for words in the candidate translation compared to the reference translations, as described in [7]. In particular, the n-gram precision is computed by taking the count of each n-gram in the reference translation, and clipping this count by the maximum number of times that the n-gram appears in a reference translation. This clipped count is divided by the unclipped count in the reference translation to produce a score.

For example, a candidate caption of "a yellow yellow yellow bus" compared to a reference translation of "a yellow bus" would have a unigram precision of $1/3$ for the word yellow. A weighted average is taken over n-grams.

The METEOR metric was introduced [[8]] to address criticisms of poor correlation between BLEU score and human evaluation, and this metric is also reported in the Results section. A python script is provided by the creators of the MSCOCO dataset for evaluation of these scores.

4.4 Qualitative Analysis of Results and Attention Mechanism

4.4.1 Representative Captions

In this section, a number of representative captions generated by the attention model are shown, with varying levels of success in idenitfying image features and correctly describing the scene. It is rare to find captions which are entirely unrelated to the image; however, a recurring issue is that rare subtypes of object classes (e.g. an alpaca or an ethnic food item) are often mislabeled as subtypes more commonly seen in the training data (e.g. a cow or a hamburger).

Successful

A close up of a bunch of broccoli

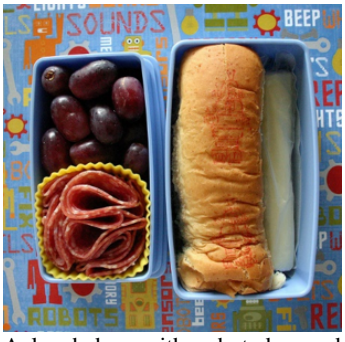
A man riding on the back of a white horse

A yellow fire hydrant in the snow

Partially Successful

A woman in a suit talking on a cell phone

A cat sitting on top of a TV on top of a chair



A lunch box with a hot dog and french fries

Unsuccessful

A woman taking a picture of a cat

A reflection of a man standing in front of a bus

A little girl sitting at a table with a plate of food

4.4.2 Analysis of Attention Mechanism

The particular attention mechanism used in this model is an "attention" over feature filters, which is then used to select the most relevant image feature vectors from a 14x14 grid. Because the network is not trained to directly output a spatial probability distribution, it was not expected that the activated image regions at a given timestep will always directly correlate with the next selected word. Indeed, a lack of a strong response to a particular output feature attention vector provides information, but does not have a simple corresponding visualization.

However, in some cases (particularly the initial timesteps when the network tends to seek large-scale "object" information, and at timesteps preceding the generation of color words), there is a visual correlation between the activated image regions and the next-word generation.

In Fig. 4, example images are shown with brightness proportional to the probability distribution over the image locations, p_v, used to generate w_t. In Fig. 5. a visualization of feature vector activations corresponding to $mask_t$ at each timestep during caption generation is shown for the same two images.

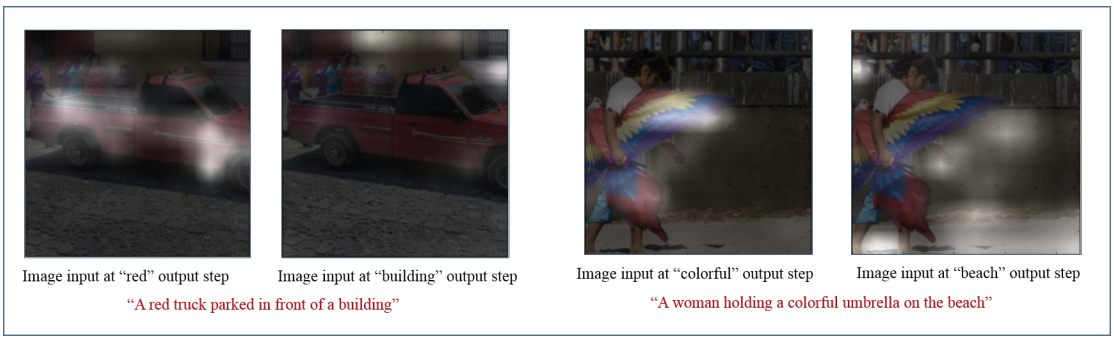


Figure 4: Activated regions of selected images from the validation dataset at particular input time-steps

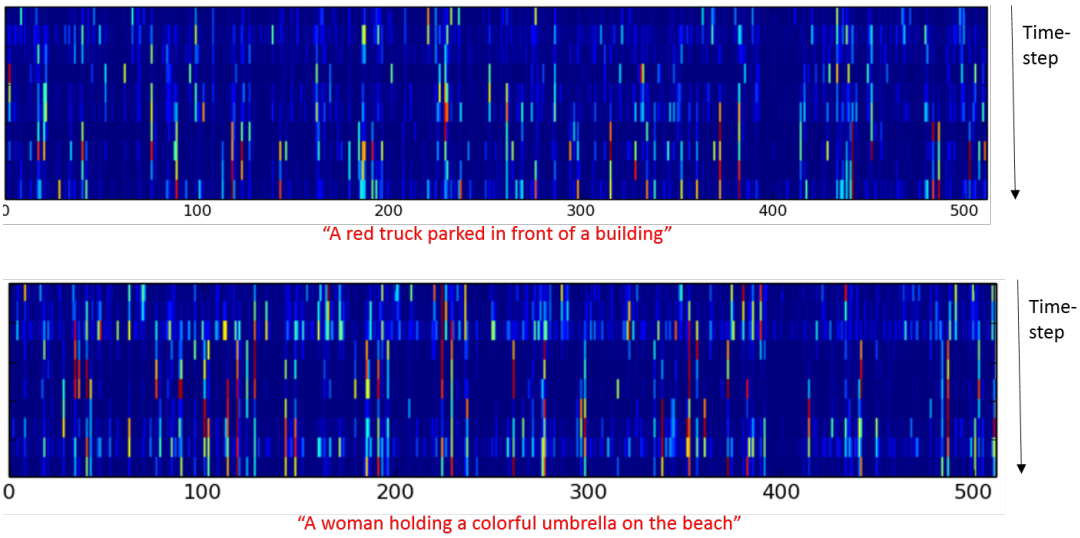


Figure 5: Activated indices of the feature mask output at each timestep for two selected images (blue indicates activation near 0, and red indicates activation near 1)

4.5 Quantitative Results

The BLEU scores and METEOR score for the models used in this work (LSTM network with attention mechanism, and baseline LSTM with no attention) are shown in Table 6, as well as corresponding scores reported in previous publications which used the MSCOCO dataset. Ideally, human evaluation of caption accuracy would also be included, due to the imperfect correlation between human scoring and the automated metrics. Indeed, as shown in the table, a source (of unvalidated origin) which created fully human-generated captions for the MSCOCO validation images and submitted these captions to the evaluation server received BLEU scores only marginally better than the LSTM baseline model in this work.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
This work (best)	68.2	**50.7**	**37.1**	**27.4**	23.1
This work (base)	64.7	48.0	31.8	22.1	21.0
Xu et. al, Hard Attention (2015)	**71.8**	50.4	35.7	25.0	23.04
Xu et. al, Soft Attention (2015)	70.7	49.3	34.4	24.3	**23.9**
Karpathy & Li (2014)	64.2	45.1	30.4	20.3	—
Human	66.3	46.9	31.1	21.7	25.2

Figure 6: Evaluation metrics for the MSCOCO dataset from this work and a selection of related previous works

Some small-scale analysis was conducted on the baseline LSTM and attention model to better quantify the specific benefits of adding an attention mechanism to the caption generation task, with results shown in Table 7. In this analysis, three word group categories were identified: color words (i.e. blue, green, red, etc.), texture words (wet, bumpy, smooth, rough, dry), and prepositional words or phrases (behind, on top, under, above). For each category, a sampling of 50 image/caption

pairs which contained words from the given category were taken from both model outputs on the MSCOCO test split. The correct usage of the word in the context of the image was then subject to human evaluation. Results showed that the attention model had better performance in correct usage of color words, slightly better performance with texture words, and no change in performance with prepositional phrases. Given that the nature of the attention mechanism was to focus on feature subtypes rather than image regions, a significant performance improvement in prepositional phrase accuracy was not expected.

Test	Baseline Model			Attention Model		
Color words	37%	32%	31%	77%	15%	8%
Texture words	59%		41%	64%		36%
Prepositional Phrases	71%		29%	69%		31%

Figure 7: Comparison of standard LSTM model and attention model for accuracy in three different sub-tasks. Green indicates the percentage of cases with correct usage, and red the percentage with incorrect word usage given the image. For the color task, a "partially correct" score was given if two color descriptors were used and only one was correct.

A downside of the Beam Search method, which was used to generate image/caption pairs before final evaluation, is that it leads to a limited usage of the available vocabulary and limited variety in generated sentences. This was also seen in Ref. [3]. Despite the vocabulary size of over 13,000 words, only 1342 words are present in the captions generated for the 40,000-image validation split by the attention model. The frequencies of the most common words seen in the training data, and generated for the validation images, are shown in Fig. 8. The word "a" is omitted from the chart but is by far the most common, composing 16% of the words in the training data and 24% of the words in the generated captions.

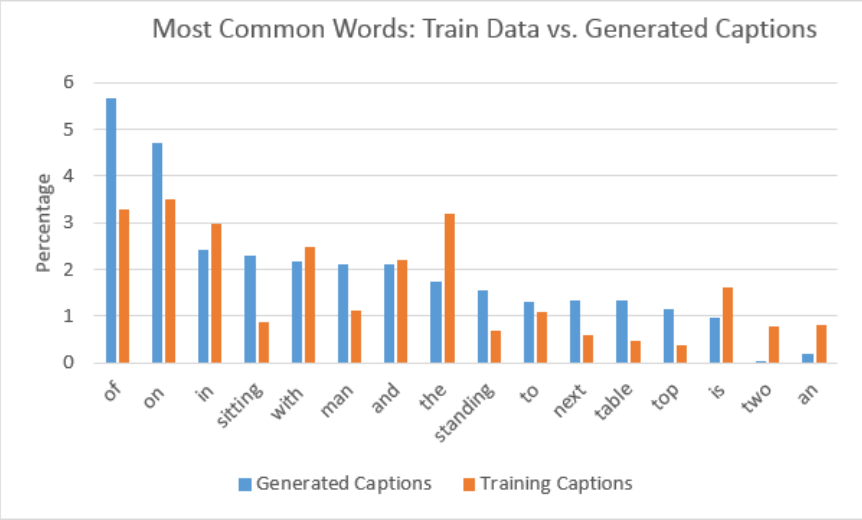


Figure 8: Relative frequencies of most common words in training data and generated captions

5 Conclusion

The LSTM-based language model with an attention mechanism introduced in this work receives evaluation scores which rival or exceed (in the case of BLEU-n score) those found in similar previous works. However, within the limitations of the MSCOCO dataset and combined CNN/RNN model approach, it is difficult to continue to make large gains in performance. In order to proceed with more accurate and descriptive caption generation that draws from a larger vocabulary, much more training data will be needed. It would be useful to have a training dataset which contains more verbose, potentially multi-sentence captions, and is filtered for spelling and syntax errors. In addition, for more informative captions which describe relative object locations within an image, it is likely that a specialized convolutional network will be required rather than the standard AlexNet or VGGNet. For example, an image network trained for instance segmentation could be used to create a small "bag of phrases" for the features found within the image and their relative locations, which is then used by the language model to produce a reasonable sentence. While the existing model already achieves high performance, it will be interesting to see this model improve as advances are made toward better visual segmentation models, better language models, and better training datasets.

References

[1] Karpathy, Andreq and Li, Fei-Fei *Deep visual-semantic alignments for generating image descriptions*. *arXiv:* 1412.2306, December 2014.

[2] Mao, Junhua, Xu, Wei, Yang, Yi, Wang, Jiang and Yuille, Alan *Deep captioning with multimodal recurrent neural networks*. *arXiv:* 1412.6632, December 2014.

[3] Vinyals, Oriol, Toshev, Alexander, Bengio, Samy and Erhan, Dumitru *Show and tell: A neural image caption generator*. *arXiv:* 1411.4555, November 2014.

[4] Xu, Kelvin et al. *Show, attend and tell: neural image caption generation with visual attention*. *arXiv:* 1502.03044, February 2015.

[5] Simonyan, Krren and Zisserman, Andrew. *Very deep convolutional networks for large-scale image recognition*. *arXiv:* 1409.1556, September 2014.

[6] Lin, Tsung-Yi et al. *Microsoft COCO: Common Objects in Context*. *arXiv:* 1405.0312, May 2014.

[7] Papineni, Kishore, Roukos, Salim, Ward, Todd and Zhu, Wei-Jing. *BLEU: a method for automatic evaluation of machine translation*. ACL Conference Paper: July 2002.

[8] Banerjee, Satanjeev and Lavie, Alon. *METEOR: An automatic metric for MT evaluation with improved correlation with human judgments*. ACL Conference Paper: June 2005.

[9] Zeremba, W., Sutskeverr, I and Vinyals, O. *Recurrent neural network regularization*. *arXiv:* 1409.2329, 2014.