Data Augmentation with Seq2Seq Models

Jason L. Granstedt

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Masters of Science
in
Electrical Engineering

Dhruv Batra, Chair
William T. Baumann
Bert Huang

February 16, 2017
Blacksburg, Virginia

Keywords: Data Augmentation, Seq2Seq, Diverse Beam Search, VQA

Data Augmentation with Seq2Seq Models

Jason L. Granstedt

ABSTRACT

Paraphrase sparsity is an issue that complicates the training process of question answering systems: syntactically diverse but semantically equivalent sentences can have significant disparities in predicted output probabilities. We propose a method for generating an augmented paraphrase corpus for the visual question answering system to make it more robust to paraphrases. This corpus is generated by concatenating two sequence to sequence translation models. We produce candidate paraphrases using diverse beam search and evaluate the results on the standard VQA validation set.

Our approach results in a significantly expanded training dataset and vocabulary size, but has slightly worse performance when tested on the validation split. Although not as fruitful as we had hoped, our work highlights additional avenues for investigation into selecting more optimal model parameters and the development of a more sophisticated paraphrase filtering algorithm. The primary contribution of this work is the demonstration that decent paraphrases can be generated from sequence to sequence models and the development of a pipeline for developing an augmented dataset.

Data Augmentation with Seq2Seq Models

Jason L. Granstedt

GENERAL AUDIENCE ABSTRACT

For a machine, processing language is hard. All possible combinations of words in a language far exceed a computer's ability to directly memorize them. Thus, generalizing language into a form that a computer can reason with is necessary for a machine to understand raw human input. Various advancements in machine learning have been particularly impressive in this regard. However, they require a corpus, or a body of information, in order to learn. Collecting this corpus is typically expensive and time consuming, and does not necessarily contain all of the information that a system would need to know - the machine would not know how to handle a word that it has never seen before, for example.

This thesis examines the possibility of using a large, general corpus to expand the vocabulary size of a specialized corpus in order to improve performance on a specific task. We use Seq2Seq models, a recent development in neural networks that has seen great success in translation tasks to do so. The Seq2Seq model is trained on the general corpus to learn the language and then applied to the specialized corpus to generate paraphrases similar to the format in the specialized corpus. We were able to significantly expand the volume and vocabulary size of the specialized corpus via this approach, we have demonstrated that decent paraphrases can be generated from Seq2Seq models, and we developed a pipeline for augmenting other specialized datasets.

Acknowledgments

I would first like to thank my thesis adviser, Dr. Dhruv Batra, of the Department of Electrical Engineering at Virginia Tech. His enthusiasm for machine learning and the access to server resources were instrumental to this project. I also deeply appreciate the creative ideas he came up with when unexpected challenges arose. He consistently allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

I would also like to acknowledge the other two members of my thesis committee, Dr. William Baumann and Dr. Bert Huang, for their support and mentorship throughout my education. Dr. Huang introduced me to the field of machine learning and ignited my passion for the field, and Dr. Baumann has supported my creative development throughout my undergraduate education in both the classroom and independent studies.

I would also like to thank Dr. Troy Nolan for his assistance in proofreading and editing. I am grateful for his valuable comments on this thesis.

Finally, I would like to thank my family for their encouragement throughout my years of study and the process of researching and writing this thesis. This accomplishment would not have been possible without their unfailing support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A question answering (QA) system is designed to take a free query as an input and output a valid answer to the posed question. The visual question answering (VQA) model combines a conventional question answering system with images to form a system capable of addressing free queries about the contents of an image. However, the model suffers from a data sparsity problem when dealing with paraphrases. Multiple syntactically different but semantically equivalent sentences yield significantly different answers. We propose to solve this issue by augmenting the VQA dataset with machine-generated paraphrases to make it more robust.

Paraphrase generation has been traditionally performed via statistical machine translation (SMT), which involves learning paraphrases as statistical distributions from a corpus. For this project, we consider the paraphrase generation problem as a translation problem. This view of the problem allows us to bring in modern translation techniques. We are primarily concerned with a variant of recurrent neural networks (RNNs) known as sequence to sequence (seq2seq) models that have improved results in the translation task over SMT.

While the translation view of paraphrase generation is not new (generating paraphrases with a bilingual corpus is known as "pivoting"), using neural networks to perform the task is a recent development - primarily because multiple paraphrases are desired for a given input phrase and traditional beam search yields very similar sentences. A recent modification to sampling neural networks known as diverse beam search allows us to diversify the generated sentences. We seek to combine the "pivoting" concept of generating paraphrases from these traditional SMT approaches with modern advancements in translation technology to develop a more efficient approach for generating paraphrases.

Our approach involves training two neural network based translation models, one from English to French and the other from French to English. These models are concatenated to generate English paraphrases by pivoting off of the French language. We then decode results from these models with diverse beam search to generate an augmented corpus of diverse paraphrases. The VQA model is then trained with this augmented data in an attempt to improve its

robustness to sentence variation and it is evaluated on the traditional VQA validation set to observe the accuracy.

Chapter 2 provides background on statistical machine translation, pivoting, sequence to sequence models, and the question answering problem. Chapter 3 details our approach for generating and sampling the fre2eng and eng2fre seq2seq models to produce the paraphrases and the results of augmenting the VQA dataset. Finally, Chapter 4 contains our conclusions and analysis of the results.

Chapter 2

Literature Review

The related works for this project are categorized into three major categories: statistical machine translation (SMT), sequence to sequence models and question answering.

2.1 Statistical Machine Translation

SMT is the traditional way paraphrases are generated. Originally proposed by Weaver [2] and implemented by Brown et al. [3], the principle idea is that paraphrases can be learned as a probability distribution from a corpus of data. The variations of SMT methods are primarily dependent on the corpus used to learn the statistical relations. The three most common avenues of research include a large monolingual corpus [4], a parallel bilingual corpus [5], and a parallel monolingual corpus [6].

The major drawback of even an optimized SMT system is its scaling: the per-phrase cost of the method is $O(d[mcc + s + |V| + min(k, \log(d) + \log|T|])$ where d is the desired number of paraphrases, V is the number of unique tokens, and T is the size of the corpus [4]. The scaling with both the number of unique tokens and the size of the paraphrase corpus makes the SMT approach inefficient compared to the massive datasets available today [7]. Furthermore, SMT approaches result in little syntactic diversity - although they are apt at replacing phrases that correspond, they are ill-suited to rearranging sentence structure [5] [8]. We attempt a different approach to increase the diversity of our generated paraphrases.

2.2 Sequence to Sequence Models

The concept of a sequence to sequence (seq2seq) model was originally proposed by Cho et. al [9] and Sutskever et. al [10]. A seq2seq model consists of two concatenated recurrent neural

networks (RNNs). The first RNN, the encoder, reads the source sentence and encodes it into a vector. The second RNN, the decoder, takes the encoded sentence as input and outputs the translation. The basic model was extended to use multi-layer cells (specifically LSTMs) by Sutskever et. al [10].

One of the early issues with seq2seq models was processing large sentences, especially those that are longer than the training corpus [9]. Bahdanau et. al addressed this problem by allowing the encoding vector to be of variable size and adding an attention mechanism [11]. The result is a model that can align and train jointly, predicting each target word based on the vectors associated with source positions and previously generated target words. Seq2seq models are typically used for the translation task [9] [10] [11], but they have been extended to give state-of-the-art performance in other tasks such as syntactic constituency parsing [12]. We use a modification of the translation problem to generate paraphrases for our augmented dataset.

2.3 Question Answering

The issue of paraphrases frequently appears in the question answering (QA) domain, where a free query is mapped to a form suitable for submission to an external knowledge base. Traditionally, this has been done with logical form embeddings of input sentences. Yang et al. provides an excellent overview and comparison of the different approaches to embedding sentences and mining relational rules [13]. New advances such as query graph mapping, proposed by Yih et. al, convert a free query into a graph suitable for submitting a query to a knowledge base [14]. These query graphs encode objects and their relationships to one another independent of syntax, making them resilient to paraphrases. Both of these approaches yielded state-of-the-art results.

A paper by Shashi et al. extends the work of Yih et al. by generating multiple paraphrases of the the input sentence before generating multiple query graphs, with the intuition that at least one query graph will map to a knowledge-base query [8]. The multiple paraphrases are generated via a learned graphical walk to produce syntactically distinct but semantically identical paraphrases. Furthermore, the structure of the produced paraphrases was found to have significantly more diversity than those generated via the SMT model by Koehn et al. [15], varying in sentence form rather than just syntax [8].

While the graph-based approaches are promising and applicable to this problem [14] [8], the code is not yet publicly available. A free sentence encoder called sent2vec is provided by Microsoft at [16], which implements the deep structure semantic model (DSSM) proposed by Huang et al. [17]. This encoder maps short text strings to a low dimensional feature vector space where similarity between sentences is computed using cosine similarity. The implementation by Microsoft also supports the updated DSSM model with convolutional-pooling structure (CDSSM) proposed by Shen et al. [18] and Gao et al. [19]. Shen et al.

found that that using a convolutional structure resulted in a significant performance increases for the Web document ranking task [18], while Gao et al. [19] demonstrated its effectiveness on addressing the data sparsity problem for the phrase-based statistical machine translation (SMT) system proposed by Koehn et al.[15]. Our approach augments the VQA dataset with machine-generated paraphrases to improve the robustness of the model [20].

2.4 Beam Search

In recent years the popularity of RNNs and LSTM models has expanded to make them the standard choice for modeling time series data in a variety of applications. Problems in this domain include solving the Maximum a Posterori problem; that is, finding the most likely output sequence for any given input. Calculating the exact inference is NP-hard in the general case, so heuristic algorithms such as beam search that take the top-K results in a greedy fashion at each time step are commonly employed. However, solutions reached with beam search give generic results [21] [22]. An improvement on the base algorithm called diverse beam search (DBS) demonstrates remarkable improvement by penalizing the selected values for future considerations, leading to a wider variety in the output [23]. In our problem, we use DBS to diversify the sentences generated from sampling the seq2seq models to produce multiple diverse paraphrases from a single source sentence.

Chapter 3

Approach and Discussion

The problem we consider is generating paraphrases for an existing language corpus to augment a visual question answering system (VQA). A question answering (QA) system is designed to take a free query as an input and output a valid answer to the posed question. The visual question answering model combines a conventional question answering system with images to form a system capable of addressing free queries about the contents of an image [20]. However, the model suffers from a data sparsity problem when dealing with paraphrases. This issue appears when dealing with both syntax (demonstrated in Figure 3.1) and diction (contained in Figure 3.2).

Paraphrases are syntactically distinct but semantically similar sentences. Since there is a loose mapping between different ways of saying the same thing, small syntactic differences tend to create large differences in semantically similar statements [19].

Many of the more traditional QA systems also suffer from the paraphrase sparsity issue, and there is a good deal of active research to reduce the detrimental effect paraphrases have on accuracy [8] [13] [14]. However, a paraphrase corpus is required to evaluate the effectiveness of any implemented method.

A closely related field to QA systems is machine translation. These models take a native sentence as input and attempt to produce a semantically identical sentence in a different language [9]. The change of language forces a different syntax onto the statement. Using a foreign language to determine the paraphrasal relationships between words of the language of interest is known as pivoting [24].

In order to increase the robustness of the system to paraphrases, we now propose a method to adapt advancements in translation to generate a paraphrase corpus suitable for use with the VQA model. Two neural network based translation models, one from English to French and the other from French to English, are concatenated to generate English paraphrases by pivoting off of the French language. This is the first time neural networks have been used to generate paraphrases; previous work focused on statistical machine translation approaches (a

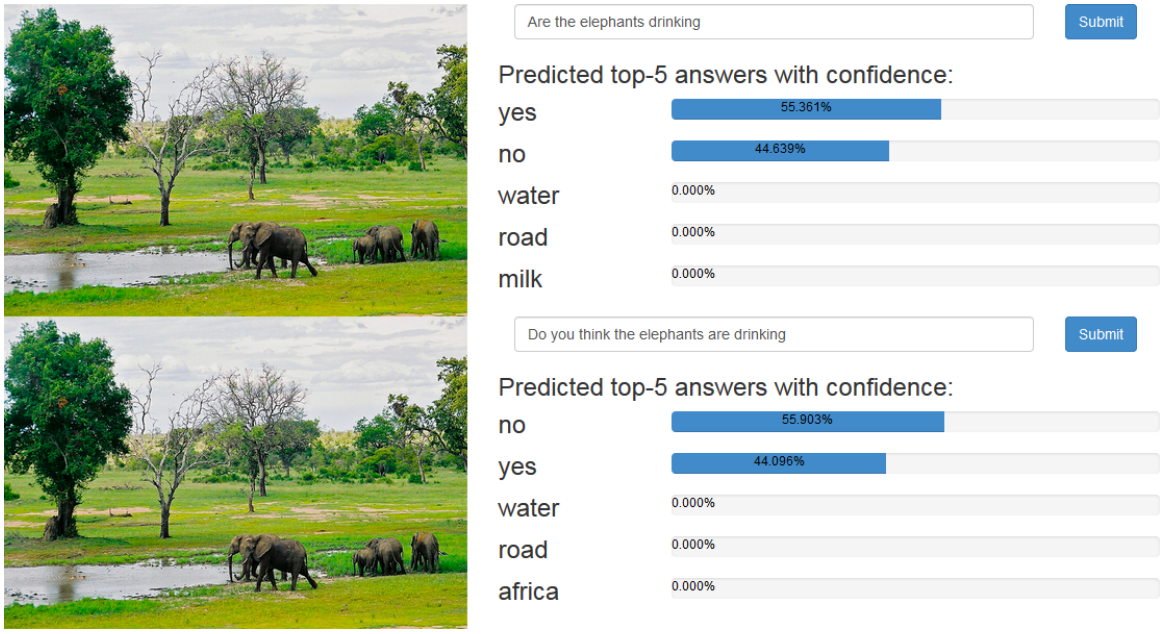

Figure 3.1: Two sample outputs from the baseline VQA system that show an answer flipping from syntax.

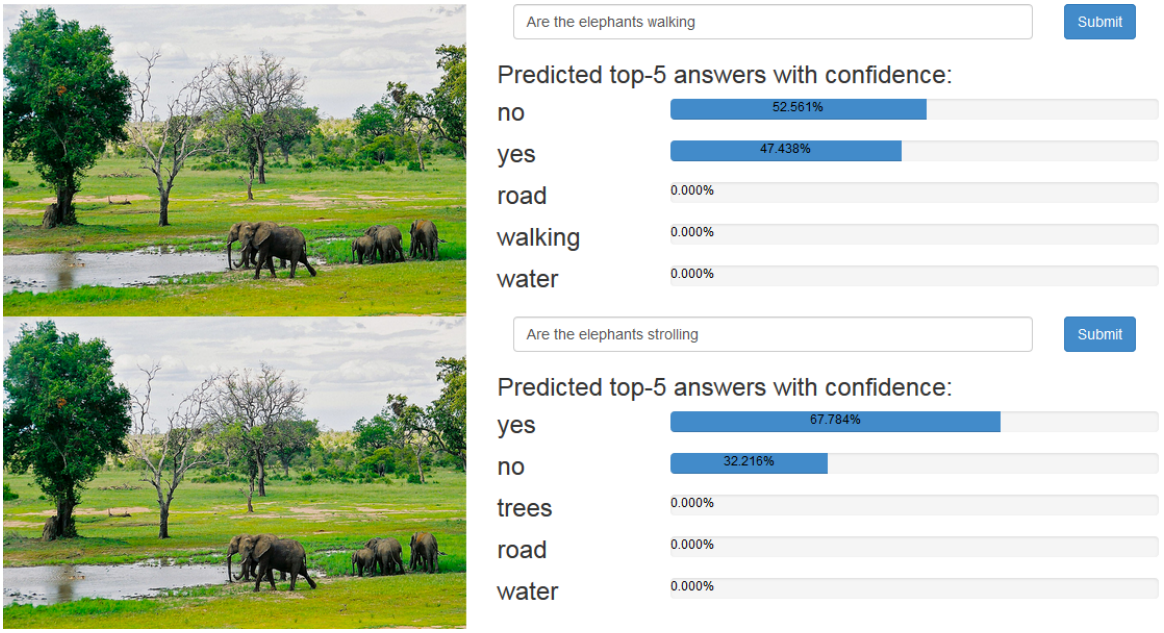

Figure 3.2: Two sample outputs from the baseline VQA system that show an answer flipping from word choice.

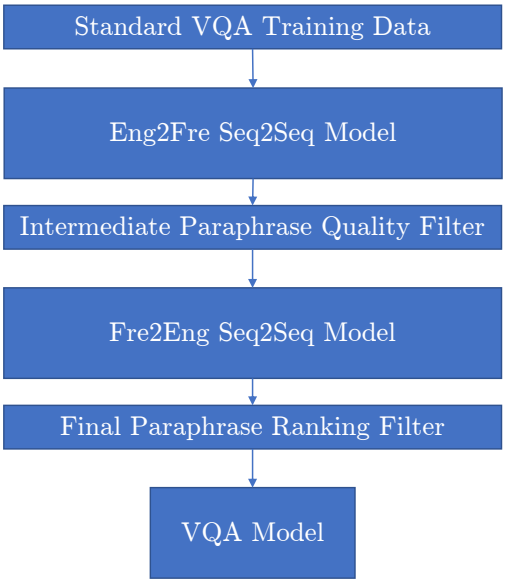

Figure 3.3: Overall pipeline of the data augmentation approach.

good summary of which by Marton are at [4]). We seek to combine the "pivoting" concept of generating paraphrases from these traditional SMT approaches with modern advancements in translation technology to develop a more efficient approach for generating paraphrases.

An overall pipeline of our approach is in Figure 3.3. Each of the component elements are described in their respective sections below.

3.1 Seq2Seq Models

The seq2seq model is selected for this task because it has constant-time performance with respect to the size of the corpus during runtime. This allows us to use a rich corpus to train the model without having the drawback of a resulting slowdown when the model is used, as would be the case with SMT methods. The translation model implemented for this project is the seq2seq model with attention, as proposed by [11]. A diagram for the model's structure is in Figure 3.4.

The dataset used to train the model is the WMT'15 French-English aligned corpus [7]. The training corpus contains approximately 22 million sentences. One of the key design parameters for our model is the vocabulary size. If the vocabulary size is too small, then many unknown tokens will appear in the file outputs and degrade the results. However, increasing the amount of tokens increases the model complexity and thus increases the training time. Sparsity also becomes an issue if too many tokens are considered; if the additional parameters of the model do not have sufficient training data, then they will not be selected even when appropriate. The decision is highly dependent on the training dataset.

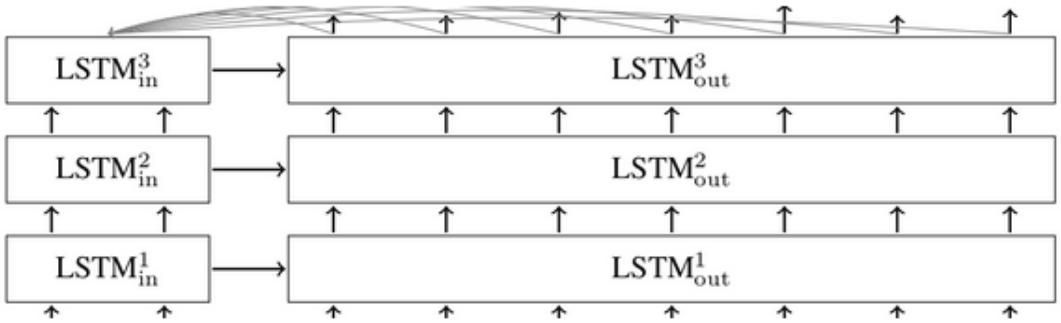

Figure 3.4: The seq2seq model used for each of the translation models. This is a three-layer seq2seq network with an attention mechanism and LSTM cells[1].

To determine the best parameters for our model, we compare the WMT dataset to the VQA vocabulary, our planned target output. We examined both the coverage and the volume gained per 10,000 tokens included, where "volume" denotes the quantity of words covered in the dataset and "coverage" encapsulates the number of unique words included. The results of our token sweep is included in Figure 3.5, with a cumulative view of the results in Figure 3.6.

The majority of the volume seems to be represented quickly, with over 85% of the data lying within the first 10,000 tokens. However, further increases to the token size result in diminishing returns. Coverage increases more slowly but has greater returns for an expanded token size. We chose a vocabulary size of 100,000 tokens because of the large size of the WMT dataset and the nature of our task: to maximize the number of questions we can completely represent. This design choice gives us a coverage of 70.5% and a volume of 98.8%.

We use NLTK's tokenizer to process the data [25] and select 3 LSTM layers with a size of 1024 units each with a final attention layer as our model parameters. The data is encoded with a bidirectional encoder and summed to produce the final state. The translation models are used after going through one epoch of the data. The models are then trained using the opennmt framework [26]. Training is done on a K40 Nvidia GPU and takes approximately 93 gpu-hours for each model. The convergence graph is in Figure 3.7. The eng2fre model has a validation perplexity of 9.42 and the fre2eng model has a final validation perplexity of 7.98.

After the English to French (Eng2Fre) and French to English (Fre2Eng) models are individually trained, they are concatenated together to form the full model. The training set of the VQA corpus is then processed by the model to generate augmented training data.

[1]The image in Figure 3.4 is reproduced from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License.

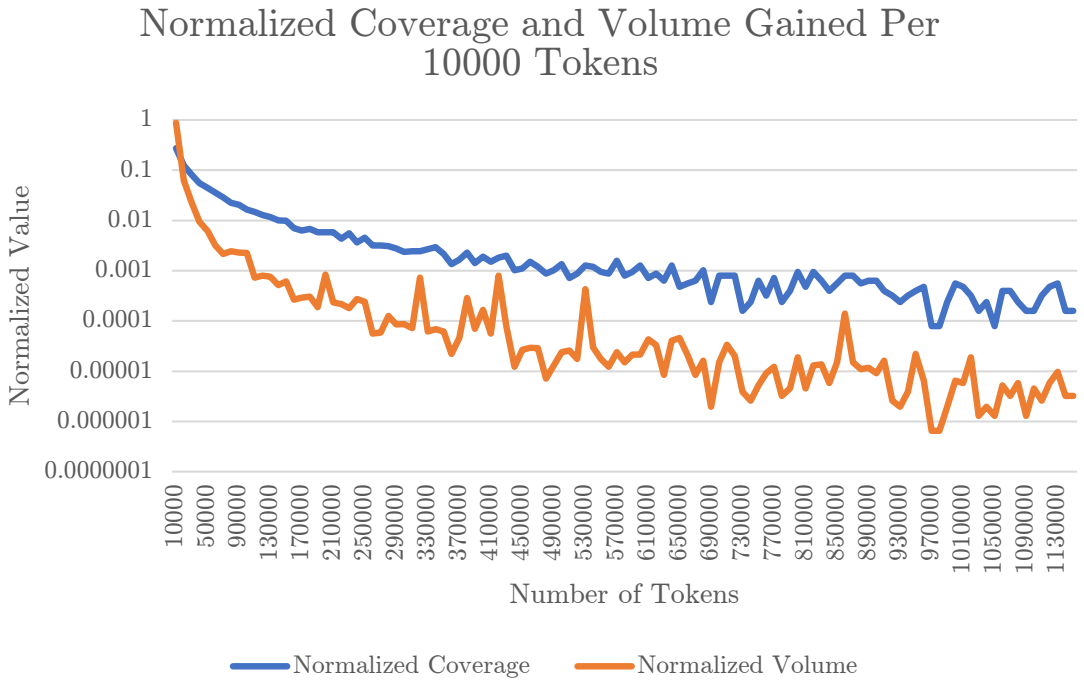

Figure 3.5: Amount of coverage and volume gained on the VQA dataset per 10,000 tokens. Coverage represents the percentage of unique words included whereas volume represents the total percentage of the dataset covered.

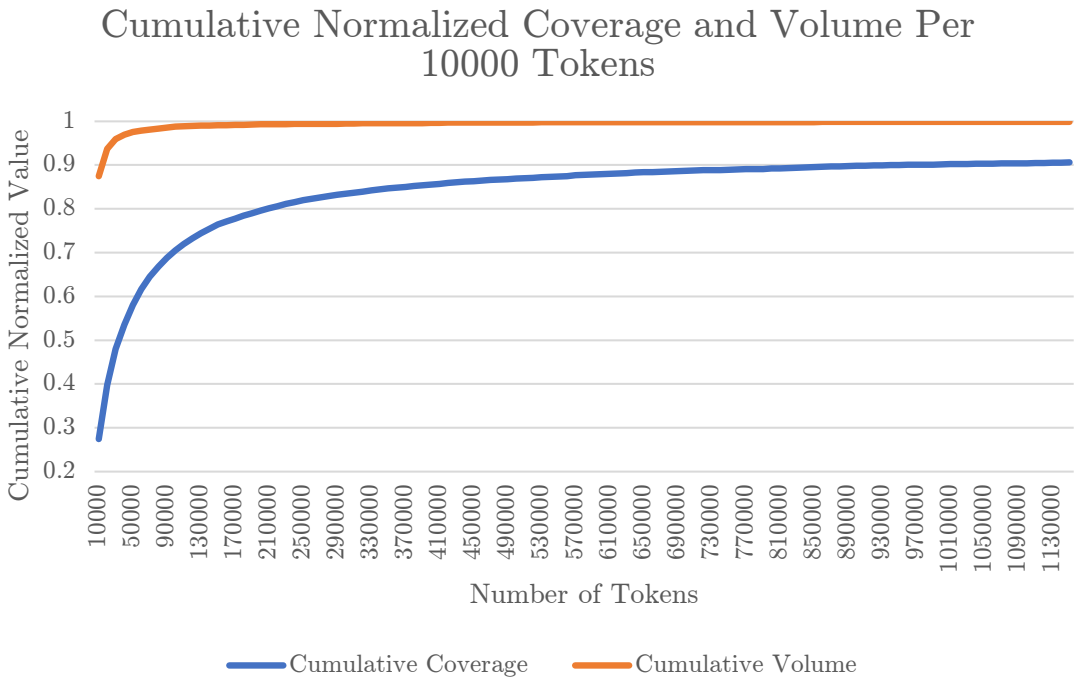

Figure 3.6: Cumulative coverage and volume per 10,000 tokens on the VQA dataset. Coverage represents the percentage of unique words included whereas volume represents the total percentage of the dataset covered.

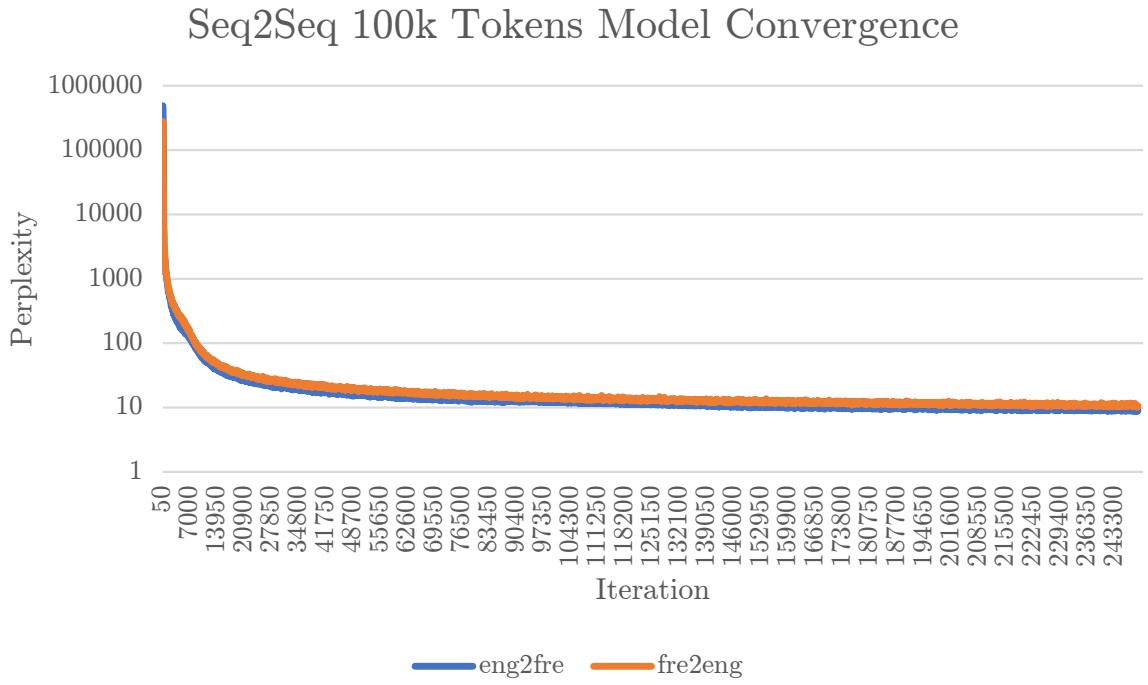

Figure 3.7: Convergence graph for the two seq2seq models trained. The eng2fre model had a validation perplexity of 9.42 and the fre2eng model had a final validation perplexity of 7.98.

3.2 Beam Search

The traditional solution to generate multiple hypothesis for a single input is beam search, where the top N results are chosen to continue at every time step. However, this tends to result in very similar output sentences [21]. Since we desire to produce diverse paraphrases in both syntax and structure, we consider the diverse beam search (DBS) algorithm [23].

Diverse beam search takes three parameters: the total beam budget N, the number of groups G, and the diversity penalty λ. Each group G has a beam size of N/G. Each time a group selects a token, further groups incur a penalty of λ to that same token for the current iteration of the beam search. This increases the chance of different tokens being selected for each group. However, even in the worst case diverse beam search is guaranteed to do at least as well as standard beam search with beam size N/G.

We experimented with several different parameters for diverse beam search to determine which one produced the best quality of paraphrases. Amongst the good paraphrases there were a significant amount of degenerate results. Two random slices of the output from diverse beam search with the parameters we selected ($B = 16$, $G = 4$, $\lambda = 0,5$) are included in Tables 3.1 and 3.2.

To mitigate the degenerate results, we need a way to select only the best. We originally tried to use the overall log likelihood of each sentence to determine the best results, but found that the likelihood did not necessarily correspond to good paraphrases. Our solution is to generate several paraphrases and then prune them using a heuristic based on the most commonly observed errors.

The errors from the DBS results can be loosely categorized as degradation (multiple chaining <unk>tokens), truncation (occurs when the sentence ends after only a few words), punctuation errors (multiple terminating punctuation symbols at the end of a sentence; an additional example is in Table 3.3), and inappropriate symbols (tokens such as <\s>and <s>are special tokens used by the seq2seq model to denote the start and end of sentences and should not appear as tokens in an output sentence).

Additionally, we observed an unusual behavior from the fre2eng seq2seq model. Unknown tokens are selected when a question mark appears at the end of a sentence while the same sentence with no question mark at the end yields a better translation. Tables 3.4 and 3.5 contain a detailed breakdown of the probabilities for the tokens in question at every beam search iteration for a sample sentence.

The errors introduced by the addition of the question mark token are intrinsic to how the model was trained, and the change in previous iterations is only made possible via the bidirectional encoder. This issue is solved by running a postprocessing script that simply removes any question marks from the ends of sentences before passing them on to the fre2eng model and restores the question marks after the final paraphrases are selected.

Table 3.1: Randomly selected subset of DBS results for another VQA question. The parameters that produced these sentences were $B = 16$, $G = 4$, and $\lambda = 0.5$. The lines in the table seperate different groups' results.

Source sentence: Is this a grocery store?
Is it an <unk>
Is this
Is it
Is it a
Do you think this is a grocery store
Do you think it is a grocery store
Do you think this is a
Do you think this is a grocery
Do you think this is a grocery store <unk>
Do you think it is a grocery store <unk>
Do you think this is a store store <unk>
Do you think it is a grocery store <unk>
<unk><unk><unk><unk><unk><unk><unk><unk><unk><unk><unk>
<unk><unk><unk><unk><unk><unk><unk><unk><unk>
<unk><unk><unk><unk><unk><unk><unk><unk><unk><unk>
<unk><unk><unk><unk><unk><unk><unk><unk><unk><unk>
Is it a store store <unk><unk><unk><unk>
Is it a store store <unk><unk><unk><unk><unk>
Is it a store store <unk><unk><unk><unk><unk><unk>
Is it a store store <unk><unk><unk><unk><unk><unk><unk>
Is it an Grocery Store
Is it a grocery store
Is it an grocery store
Is it an
Is it an independent grocery store <unk><unk>
Is it an independent grocery store <unk>
Is it an independent grocery <unk><unk>
Is it an independent grocery store <unk><unk>
Is it a grocery store <unk>
Is it a grocery store <unk><unk>
Is it a grocery store <unk><unk><unk>
Is it a grocery store </s>

Table 3.2: Randomly selected subset of DBS results for another VQA question. The parameters that produced these sentences were $B = 16$, $G = 4$, and $\lambda = 0.5$. The lines in the table separate different groups' results.

Source sentence: Are these items for sale ?
<unk>these <unk>; ; <unk>these <unk>; <unk> <unk>these <unk>; ; <unk> <unk>these <unk>;
Do you think these articles are intended for sale Do you think these articles are intended for sale <unk> Do you think these items are intended for sale <unk> Do you think these articles are intended for sale <unk>
Are these items intended for sale <unk>. . . . . . Are these items intended for sale <unk><unk><unk><unk><unk> Are these items intended for sale <unk><unk><unk><unk><unk><unk> Are these items intended for sale <unk><unk><unk><unk><unk><unk><unk>
Do these articles are intended for Do these articles have been used for Do those articles have been intended for sale Do these articles have been
Are these items intended for sale in the <unk> Are these items intended for sale in the sale of <unk> Are these items intended for sale in the sale <unk> Are these items intended for sale in the sale
Do they want to sell Do they want to be sold Do they want to be Do they have a
Do you think these articles are intended for sale Do you think these articles are intended for sale <unk> Do you think these items are intended for sale <unk> Do you think these articles are intended for sale <unk>
<unk><unk><unk><unk><unk><unk> <unk><unk><unk><unk><unk><unk><unk> <unk><unk><unk><unk><unk><unk> <unk><unk><unk><unk><unk><unk><unk>

Table 3.3: A specific slice of two beams for a VQA question that demonstrates multiple terminating punctuation symbols. When the rest of the sentence is similar, the next highest probability token after EoS is typically a punctuation mark or unknown symbol. This produces results that add little diversity to the existing sentences. The parameters that produce these sentences are $B = 16$, $G = 4$, and $\lambda = 0.5$. The lines in the table separate different groups' results.

Source sentence: How many birds ?
How many birds are <unk>!
How many bird <unk>! <\s>
How many birds are <unk>! !
How many birds are <unk>! ?
How many birds are <unk>! !
How many birds are <unk>! <\s>
How many birds are <unk>! ?
How many birds are <unk>! ! !

Table 3.4: Token probability breakdown for beam search processing of the French sentence "Combien d'animaux peuvent être vus" (Original sentence: How many animals can be seen).

	Token Probabilities				
iter	<unk >	seen	?	EoS	Chosen Token
1	-2.7008	-15.4719	-9.6728	-6.6676	How
2	-9.8116	-16.2888	-12.1351	-11.094	many
3	-8.1694	-16.7946	-14.9014	-10.4833	animals
4	-8.0671	-13.7472	-13.8074	-9.072	can
5	-7.16	-12.7076	-16.4976	-8.2003	be
6	-3.528	-1.4248	-17.3126	-6.2692	seen
7	-3.2221	-11.5292	-6.895	-0.4692	EoS

Table 3.5: Token probability breakdown for beam search processing of the French sentence "Combien d'animaux peuvent être vus ?" (Original sentence: How many animals can be seen ?). The only change from the previous table is the addition of a space and question mark.

	Token Probabilities				
iter	<unk>	seen	?	EoS	Chosen Token
1	-3.3902	-16.0216	-9.0174	-7.4595	How
2	-9.1499	-15.9191	-12.0337	-12.1352	many
3	-6.9169	-16.065	-13.5345	-10.6123	animals
4	-7.7821	-17.075	-12.2799	-11.1319	can
5	-2.8398	-12.7931	-10.5662	-7.1098	be
6	-0.1979	-3.5217	-12.3806	-8.7105	<unk>
7	-4.6358	-14.714	-7.2668	-0.015396	EoS

One final issue with the pruning method is computation time. Since we are concatenating two seq2seq models, the total beam budget is squared. This results in a significant computational cost increase. While we do want to have a good number of paraphrases at the end to select from, several results may be pruned from the intermediate French translation step.

3.2.1 System Modifications

To improve the runtime generating paraphrases, we start by filtering out all of the candidates with illegal tokens. These tokens are reserved as special symbols and should not appear in the final result. We then check for and remove sentences with multiple punctuation marks. The information contained in these sentences is typically represented by an earlier paraphrase without the extra punctuation; the additional symbols are artifacts of the beam search process. Sentences are also filtered out if the candidate paraphrase is more than two words shorter than the original sentence. This restriction is put in place to filter out the truncations. Finally, any sentence with an unknown token density greater than one unknown per five words is removed.

Most of the candidate paraphrases filtered by the chosen criteria are indeed erroneous paraphrases. The small percentage of valid paraphrases pruned is an acceptable cost given how many total paraphrases are being generated. This intermediate filtering process reduces the number of candidate French paraphrases by approximately half, from 3,973,584 to 2,054,498.

Our goal with the final pruning filter is to maximize the diversity of the selected paraphrases while reducing the amount of errors. We implement a scoring system with many of the same characteristics of the intermediate pruning system to rank the best paraphrases. The starting

diversity score is the set of words that form the sentence. This score is then subtracted by several penalties. The overlap penalty is the intersection between the set of the paraphrase and the original English sentence. A length penalty is assigned if the candidate paraphrase is more than 2 words shorter than the original sentence to mitigate the impact of truncated paraphrases. The candidate also receives a penalty for every unknown token that it contains. Finally, a penalty is applied for any illegal characters and multiple punctuation symbols. For this experiment, the coefficient per illegal token violation is set to ten. No instances of multiple punctuation or illegal characters are found in the final results. The top three candidate paraphrases are selected from the entire batch, provided that their scores are greater than zero. This second filter narrowed the results from 32,871,968 to 879,104. We call the resulting dataset from this above process "augmented".

We also experiment with the settings of the second filter to observe the effects that unknowns have on the final accuracy. To do so, we change the penalty on the unknown tokens to be equivalent to that for illegal characters. We also force the selected paraphrases to be diverse by changing the overlap penalty to include words from previously selected paraphrases for the current question. These modifications narrowed the results from 32,871,968 to 861,584. Curiously, the number of final paraphrases only dropped by about 20,000 despite the much harsher penalties. We call the dataset from this process "diverse".

The entire DBS generation process took 68 GPU-hours on K40 GPUs. Sample results for the augmented dataset are contained in Table 3.6 and results from the diverse dataset are in Table 3.7. The same sample results for a standard beam search process with the same filters applied is located in Table 3.8. Both DBS results outperform standard beam search in terms of included vocabulary and accurate paraphrases. However, there is still room for a significant amount of improvement in terms of overall paraphrase quality.

One of the major issues encountered when selecting paraphrases is the tradeoff between diversity and accuracy. Changing one word can completely change the meaning of a sentence, with the most famous example being negation (the addition or removal of "not"). Although this simple filtering approach reduces a large chunk of the invalid paraphrases, it cannot distinguish if a candidate is actually a valid paraphrase of the source. Implementing a more sophisticated filter will likely help to improve results in the future.

3.3 VQA Model

At the end of the diverse beam search step, we have two datasets for consideration on the VQA model: augmented and diverse. We ran both of the datasets through the original VQA model (long short-term memory model for language, convolutional neural network for images, no attention) [20] to evaluate the effects of augmenting the dataset without considering an attention component [27]. The results are compared to the baseline VQA dataset to evaluate their performance. The models were run for 150,000 iterations using the default model

Table 3.6: Sample diverse beam search results after the "augmented" filter. DBS parameters were $B = 16$, $G = 4$, and $\lambda = 0.5$.

Is this a grocery store ?
Do you think it is a grocery store?
Do you think that this is?
Do you think that this is a?
Are these items for sale ?
Do you think they are selling?
Do those articles have been intended for sale?
Do they want to sell their?
What is for sale under this tent ?
What does it mean for the sale under this?
What does it do for the sale under the?
What do we need to do for the sale under that <unk>?
Is the bird sitting on a plant ?
Are there any birds sitting on an?
Do you have any birds sitting on?
Are there any birds sitting on a plant?
What color is the keyboard ?
How are you using the keyboard as a?
How are you using the keyboard as a colour <unk>?
How are you using the keyboard as a <unk>?
How many animals can be seen ?
How many animals can they be viewed?
How many animals can be viewed?
How many animals can be considered?
What shape is the bench seat ?
Which form is the seat of the House?
How is the seat of the head of?
What form is the seat of the seat of the group?

Table 3.7: Sample diverse beam search results after the "diverse" filter. Note the lack of unknown tokens in the data and increased vocabulary size. DBS parameters were $B = 16$, $G = 4$, and $\lambda = 0.5$.

Is this a grocery store ?
Do you think it is a grocery store?
Is it an Grocery Store?
Do the grocery store are a convenience?
Are these items for sale ?
Do you think they are selling?
Do those articles have been intended for sale?
Do they want to be sold?
What is for sale under this tent ?
What do you need to do to sell?
What does it mean for the sale under this?
What is a sales under this?
Is the bird sitting on a plant ?
Are there any birds sitting on an?
Do you have any birds sitting on?
Does the birds sit on a?
What color is the keyboard ?
How are you using the keyboard as a?
Is there any colour that?
How does the keyboard form?
How many animals can be seen ?
How many animals can they be viewed?
How many animals can be considered?
How many animals can be found?
What shape is the bench seat ?
Which form is the seat of the House?
How is the seat of the head of?
What form is the seat of the seat of the group?

Table 3.8: Sample standard beam search results processed using the "augment" filter. A beam size of 16 was used to generate the base sentences. Note that the standard beam search algorithm was unable to provide any satisfactory sentences for the question "How many animals can be seen ?"

Is this a grocery store ?
Do they think it is a food UNK?
Does this mean that it is a food UNK?
Do they think this is a food UNK?
Are these items for sale ?
These articles are sold in <unk>?
These items are sold to the <unk>?
Do items are sold in the <unk>?
What is for sale under this tent ?
What does the sale in this case mean?
What does the sale of this approach mean in this <unk>?
What does the sale of this approach mean for this <unk>?
Is the bird sitting on a plant ?
Do you have any bird sitting on a <unk>?
Does the bird sitting on a plant?
Do you have a bird sitting on a <unk>?
What color is the keyboard ?
How does the keyboard colour <unk>?
What is the colour keyboard?
What is the Colour keyboard?
How many animals can be seen ?
unused slot
unused slot
unused slot
What shape is the bench seat ?
How does the place of the Bank <unk>?
How does the seat of the Bank <unk>?
What form does the Bank <unk>?

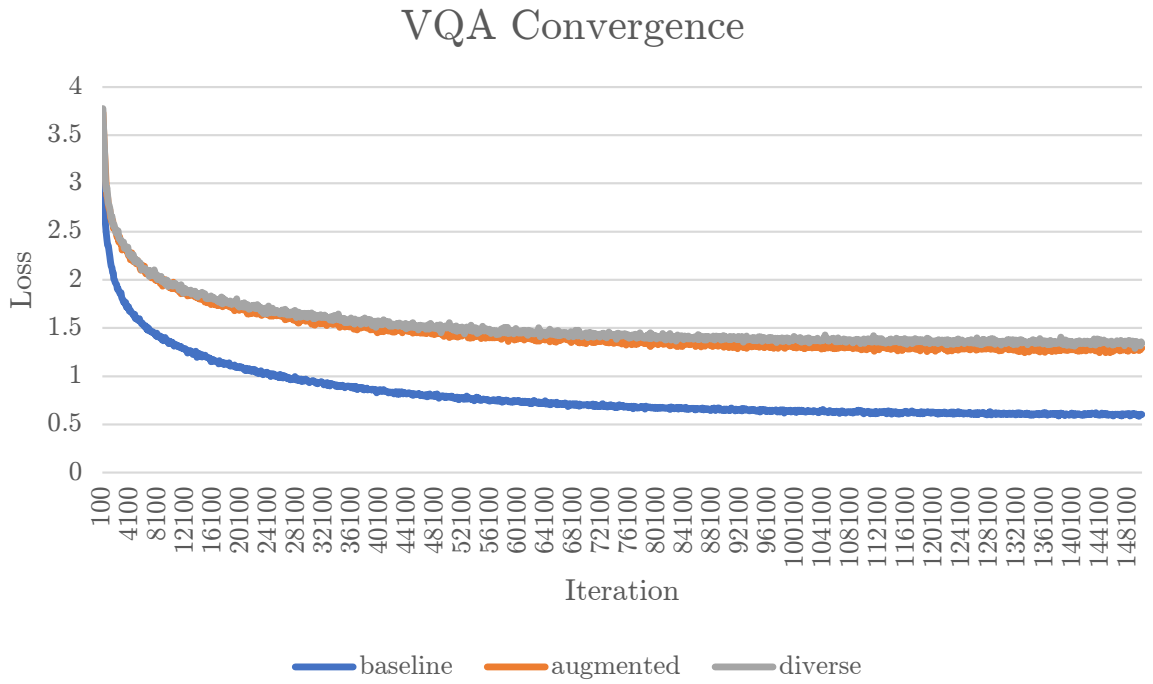

Figure 3.8: Convergence graph for the three VQA models trained.

parameters on the train/validation data split. The convergence results are included in Figure 3.8 and Table 3.9 contains a summary of the results.

Although both expanded corpora are significantly larger than the baseline VQA data in both the number of training questions and vocabulary size, the altered datasets perform slightly worse. Combined with the inferior convergence of the VQA model for the diverse datasets it appears that the additional paraphrases are not of sufficient quality to result in an improvement to the system. Both of the augmented models perform approximately the same, so it is likely the reduction in performance is due more to errors in the paraphrase generation process that resulted in semantic differences rather than proliferation of unknown

Table 3.9: Table of results for the data augmentation process. Note that the values for the number of questions are less than those reported in Section 3.2 because the VQA training code prunes some questions during its preprocessing.

	Baseline	Augmented	Diverse
Num Questions	215375	764257	748211
Vocab Size	12607	15377	15373
OpenEnded Accuracy	**54.31**	53.11	53.25
MultipleChoice Accuracy	**59.33**	58.60	58.58

tokens. Since all of the paraphrases are assigned the same answer as the original sentence, errors in the paraphrases hinder the learning process for the VQA model and result in worse performance.

Chapter 4

Conclusions

Ultimately, augmenting the VQA dataset with generated paraphrases is unsuccessful in improving the performance of the model. The lack of improvement in performance is likely due to the generated paraphrase quality, as both the number of questions and vocabulary size for the model are increased. There are a number of areas that could be altered to improve the paraphrase quality.

First, the seq2seq models did not converge as much as expected. The validation perplexity for the eng2fre and fre2eng models was 7.98 and 9.42, respectively. Previously trained 40k token models had validation perplexities around 3, performing significantly better than the 100k token models that were trained for this paper. The decrease in validation perplexity indicates that perhaps too many tokens were chosen for training the models. Choosing a more moderate number of tokens, such as 60k or 80k, may result in better performance while still sufficiently expanding the vocabulary of the VQA model.

The poor performance of the models were demonstrated when performing beam search on the concatenated models, particularly the fre2eng one. Several of the highest likelihood paraphrases were of dubious quality, especially high probabilities of strings comprised entirely of unknown tokens. The fre2eng model also had the peculiar behavior of transforming a valid token at the end of a sentence into an unknown one when a question mark was appended. This was only possible because of the bidirectional encoder, but the result reinforces the idea that the models were insufficiently trained. A study of the optimum parameters for the seq2seq models would be useful for generating higher-quality paraphrases.

We attempted to attenuate the paraphrase quality issue by developing a filtering algorithm that removed most of the degenerate paraphrases, but we were unable to determine the cases where the semantics of the sentence were changed. A more sophisticated language filter would likely greatly improve the filtering capability.

Finally, the expanded corpus and vocabulary were insufficient to improve the VQA results. However, the augmented models were not far behind the baseline results. The lower rate

of convergence on both augmented models indicates that the issue was with the training data, as discussed above. We are unable to tell at this time if the expanded vocabulary of the augmented datasets helped on the validation data performance; therefore, checking the overlap of the increased vocabulary with validation set vocabulary would indicate if any of the added words were relevant to the evaluation set. There also may be potential optimization issues from using the existing VQA code due to the increased size of the augmented datasets. Further tweaking of the VQA training code might yield better results for the existing generated paraphrases.

The primary contribution of this work is the demonstration that decent paraphrases can be generated from sequence to sequence models and the development of a pipeline for developing an augmented dataset. Future work along the lines of optimizing model parameters and improving the filtering algorithm should be considered as they will improve the quality of the generated paraphrases. With a higher-quality augmented dataset it should be possible to improve the general performance of the VQA model.

Chapter 5

Bibliography

[1] Google, "Sequence-to-sequence models." `https://www.tensorflow.org/tutorials/seq2seq/`. Accessed: 2016-12-03.

[2] W. Weaver, "Translation," *Machine Translation of Languages*, 1949.

[3] P. Brown, J. Cocke, S. D. Pietra, V. D. Pietra, F. Jelinek, R. Mercer, and P. Roossin, "A statistical approach to language translation," in *Proceedings of the 12th Conference on Computational Linguistics - Volume 1*, COLING '88, (Stroudsburg, PA, USA), pp. 71–76, Association for Computational Linguistics, 1988.

[4] Y. Marton, "Distributional phrasal paraphrase generation for statistical machine translation," *ACM Trans. Intell. Syst. Technol.*, vol. 4, pp. 39:1–39:32, July 2013.

[5] C. Callison-Burch, "Syntactic constraints on paraphrases extracted from parallel corpora," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, (Stroudsburg, PA, USA), pp. 196–205, Association for Computational Linguistics, 2008.

[6] R. Barzilay and K. R. McKeown, "Extracting paraphrases from a parallel corpus," in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, (Stroudsburg, PA, USA), pp. 50–57, Association for Computational Linguistics, 2001.

[7] "Shared task: Machine translation." `http://www.statmt.org/wmt15/translation-task.html`. Accessed: 2016-12-05.

[8] S. Narayan, S. Reddy, and S. B. Cohen, "Paraphrase generation from latent-variable pcfgs for semantic parsing," *CoRR*, vol. abs/1601.06068, 2016.

[9] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014.

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[12] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. E. Hinton, "Grammar as a foreign language," *CoRR*, vol. abs/1412.7449, 2014.

[13] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *CoRR*, vol. abs/1412.6575, 2014.

[14] S. W.-t. Yih, M.-W. Chang, X. He, and J. Gao, "Semantic parsing via staged query graph generation: Question answering with knowledge base," in *Proceedings of the Joint Conference of the 53rd Annual Meeting of the ACL and the 7th International Joint Conference on Natural Language Processing of the AFNLP*, ACL – Association for Computational Linguistics, July 2015.

[15] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, (Stroudsburg, PA, USA), pp. 48–54, Association for Computational Linguistics, 2003.

[16] "Sent2vec." `https://www.microsoft.com/en-us/download/details.aspx?id=52365`. Accessed: 2017-01-31.

[17] *Learning Deep Structured Semantic Models for Web Search using Clickthrough Data*, October 2013.

[18] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," CIKM, November 2014.

[19] J. Gao, X. He, S. W.-t. Yih, and L. Deng, "Learning continuous phrase representations for translation modeling," Association for Computational Linguistics, June 2014.

[20] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: visual question answering," *CoRR*, vol. abs/1505.00468, 2015.

[21] K. Gimpel, D. Batra, C. Dyer, and G. Shakhnarovich, "A systematic exploration of diversity in machine translation," in *In Proc. of EMNLP*, 2013.

[22] J. Li and D. Jurafsky, "Mutual information and diverse decoding improve neural machine translation," *CoRR*, vol. abs/1601.00372, 2016.

[23] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. J. Crandall, and D. Batra, "Diverse beam search: Decoding diverse solutions from neural sequence models," *CoRR*, vol. abs/1610.02424, 2016.

[24] J. Ganitkevitch, B. Van Durme, and C. Callison-Burch, "PPDB: The paraphrase database," in *Proceedings of NAACL-HLT*, (Atlanta, Georgia), pp. 758–764, Association for Computational Linguistics, June 2013.

[25] E. Loper and S. Bird, "Natural language toolkit."

[26] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "OpenNMT: Open-Source Toolkit for Neural Machine Translation," *ArXiv e-prints*.

[27] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," *CoRR*, vol. abs/1606.00061, 2016.