# Statement of Purpose

## Introduction

In the past five years, the field of natural language processing (NLP) has adopted neural network based methods for many applications. While flexible, neural network models are generally overparamaterized and take many examples to train. Especially for applications that depend on recurrent neural networks (RNNs), one should take advantage of the underlying structure when designing the network topology. With that in mind, I would like to submit my candidacy for a Ph.D. in Computer Science at Harvard University with a focus on NLP, especially in low resource settings.

## Research Interests

Rather than training deeper networks, I find the orthogonal problem of making models more data-efficient by designing biases into systems promising. Progress on this front will encourage models to be both more computationally and statistically efficient, and can be applied separately to both language understanding and generation. The reason for the increase in efficiency is as follows: an RNN processes a sequence of length $n$ in $\Theta(n)$ steps with a parallel processor, but a binary tree representation would take $\Theta(\log n)$ steps on the same parallel processor. A task-dependent tree representation would likely be a compromise between the two. In the same way that local structure is exploited in convolutional neural networks for vision and speech, one could design networks to learn the structure of natural language as well. A structure-discovering model could also lead to connections with tree-based methods from machine translation (MT). Personally, a network equivalent to an ensemble of soft tree adjoining grammars would be very satisfying.

On the same theme of using structure, there has been some recent work on generating text augmented by syntax. The appeal here is to hopefully leverage structure, possibly syntactic, to bias generation and help with imitation learning by limiting the search space during decoding. Although there is a less clear route towards an efficient network structure here, creating interesting and challenging tasks could help with promoting research in this area. One example of such a task could be learning to compile or more specifically transpile, which is the act of taking source code in one language and compiling it into source code in another language, usually at a similar level of abstraction. Applying modern encoder decoder approaches would likely take more data to complete the task successfully than if one were to leverage structure. The results would be more evaluable than natural language and the task as challenging as translation, especially when transpiling from a language that uses one paradigm into one that uses a completely different one. It is also easier to generate datasets with multiple target translations automatically. Furthermore, code generation would be a great testbed for hierarchical generation.

## Past Experience

To date, I have had the opportunity to take part in several research projects where I gained skills that will aid me in my graduate studies. I started on one such project when I joined Facebook AI Research (FAIR) in October 2015 and worked with Michael Auli on scaling FAIR's neural MT system. Given Sennrich et al.'s [3] then-recent success with byte pair encoding, we wanted to apply another technique listed in their paper on our system: keeping a shortlist of 50k unigrams and segmenting the rest into character bigrams. This resulted in a 2 BLEU score increase with our model at the time. One observation was that during decoding many words generated from subwords were misspelled. To correct for this, I introduced a lookup trie and discarded hypotheses as soon as they failed to remain on a valid path in the trie. This approach was unsuccessful since it was equivalent to giving a penalty of minus infinity to most words generated with word pieces and severely limited the size of the vocabulary. In light of this, the usage of external dictionaries to subword models could be a future research direction. Additionally, I worked on distributed training, primarily using synchronous EASGD on multiple GPUs.

For my next project I worked with Armand Joulin and Jonas Gehring on writing a new open source library in Torch (a deep learning framework) for RNNs. The goal was to provide a more flexible and performant RNN library to the community. In order to achieve this, I ensured that Cudnn baselines as well as new architectures could be constructed using almost the same interface. The torch-rnnlib library abstracts loop computations as scans so that networks can be unrolled over both time or depth, giving the user the flexibility to create networks in a layer-wise fashion or over time. As a result, it is very easy to implement conditional language models with attention-based recurrent cells, as well as stack Cudnn LSTM modules on top of said attention-based cells using the library. The library will hopefully be a welcome contribution to the open source research ecosystem.

While working on the torch-rnn library, I also co-authored a paper on ResNet normalization schemes with Wendy Shang and Kihyuk Sohn. The goal of this paper was to explore whether batch normalization was needed to stabilize ResNet training or if some other form of normalization could be used instead. I scaled the code to ensure that the modified ResNet experiments on Imagenet finished quickly and in time for submission to a conference. The paper showed that normalization propagation [1] with concatenated ReLU [4] units is competitive to batch normalization. Despite the paper's focus on a different domain than what I specialize in, it was beneficial to learn about ResNets and the different normalization strategies available since they will definitely be useful in future projects that involve training deep networks. Although, hopefully a principled approach will prevent the need for deeper models.

More recently, I have been working with Tomas Mikolov, Edouard Grave, and Armand Joulin on a simple recurrent translation system. Modern machine translation systems typically encode the source sentence, have some form of soft attention over the output of the encoder, and finally generate the target sentence one word at a time. For languages that generally have a diagonal alignment, the attention adds computational overhead. I have implemented a RNN model that consumes a source and target token simultaneously at each timestep as a starting point, as in Auli et. al. [2]. The target sentence is padded with $N$ delay tokens in order to have a lookahead of $N$ in the source sentence. If successful, this project could lead to systems that can ingest much more data given the same amount of training time and generate in real time. Since the system is limited to language pairs with largely diagonal alignments, this also leads to exciting possibilities with pre-translation source-side reordering.

### Conclusion

Given my research interests, Harvard University is the ideal location for me to pursue a Ph.D. I am especially excited by the prospect of working with one or more of Harvard University's renowned computer science faculty, including Sasha Rush and Ryan Adams. By working with leading experts in both applied and theoretical ML at Harvard University, I would have the chance to reinforce and build upon my knowledge in order to push the boundaries of NLP. In conclusion, I would be thrilled to pursue a Ph.D. at Harvard University.

### References

[1] Devansh Arpit, Yingbo Zhou, Bhargava Urala Kota, and Venu Govindaraju. Normalization propagation: A parametric technique for removing internal covariate shift in deep networks. 2016.

[2] Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks. 2013.

[3] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909, 2015.

[4] Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. Understanding and improving convolutional neural networks via concatenated rectified linear units. *CoRR*, abs/1603.05201, 2016.