SAND2018-6773C

Efficient transfer learning for neural network language models

Jacek Skryzalin*, Hamilton Link*, Jeremy Wendt*, Richard Field*, and Samuel N. Richter†
Sandia National Laboratories*, Missouri University of Science and Technology†
Email: {jskryza, helink, jdwendt, rvfield}@sandia.gov, snr359@mst.edu

***Abstract*—We apply transfer learning techniques to create topically and/or stylistically biased natural language models from small data samples, given generic long short-term memory (LSTM) language models trained on larger data sets. Although LSTM language models are powerful tools with wide-ranging applications, they require enormous amounts of data and time to train. Thus, we build general purpose language models that take advantage of large standing corpora and computational resources proactively, allowing us to build more specialized analytical tools from smaller data sets on demand. We show that it is possible to construct a language model from a small, focused corpus by first training an LSTM language model on a large corpus (e.g., the text from English Wikipedia) and then retraining only the internal transition model parameters on the smaller corpus. We also show that a single general language model can be reused through transfer learning to create many distinct special purpose language models quickly with modest amounts of data.**

I. Introduction

A neural network language model (NNLM) is a neural network designed to provide a probability distribution over word sequences in a language [1], [2]. Although generally trained to predict a term given a sequence of previous terms, NNLMs have broad applications in machine translation [3], [4], parsing [5], [6], question answering and information retrieval [7], [8], speech recognition [9], [10], text summarization [11], [12], and topic analysis and classification [13], [14].

Training an NNLM typically requires an incredibly large dataset. For example, the "one billion word benchmark" (a common dataset used to evaluate NNLMs) has a training set consisting of over 30 million sentences and over 768 million words [15]. For many applications, acquiring and cleaning a dataset of such a large size is prohibitively expensive. Intuitively, most of the information learned from such a data set is redundant from one model to the next. The general structure and patterns of a language need not be relearned for each application if a general model can be leveraged across multiple applications. This has led us to investigate transfer learning techniques for neural networks with the aim of tailoring NNLMs trained on large datasets to data for which our training set has a much more modest size.

Following [1], [2], we first train a long short-term memory (LSTM) NNLM on a large dataset (English Wikipedia), which we call the *general* dataset. We then refine our NNLM on a smaller dataset (e.g., sentences in English Wikipedia pertaining to movies), called the *biasing* dataset, by freezing the weights in the input and output layers and retraining only the model parameters which control the transitions between hidden states in the network. Although freezing and retraining has been used previously in the image understanding domain [16], [17], this is to the best of our knowledge the first application of this technique to training specialized NNLMs. The retraining phase of our technique lasts a small fraction (typically under 5%) of the time required to complete the first training phase, requires orders of magnitude less training data, and brings significant perplexity gains to language models tailored to smaller biasing datasets. That is, this is a relatively inexpensive technique which leads to significant improvements.

II. Related Work

There has been much related work in language modeling which incorporates style and content into a general language model. Although our approach is comparable, we are applying transfer techniques much more aggressively, and we are interested in techniques which might theoretically apply to a broad set of analytical tasks within the natural language processing (NLP) domain.

We note that there is much research on transfer learning in the NLP domain where the authors' approach does not involve NNLMs (e.g., Blitzer et al.'s work on part-of-speech tagging [18]). Furthermore, transfer learning is applied to neural networks in many other fields of research, most notably image representation learning [16], [17]. For the rest of this section we focus on relevant approaches to tuning NNLMs.

Motivated by the philosophy that adjacent sentences in a document should share topics and that the topic of a sentence should not change halfway through the sentence, Ji et al. incorporate context into a language model by adding a "context" vector as input to each step of a recurrent neural network in addition to a term [19]. Whereas Ji et al. train internal parameters pertaining to context and sentence terms simultaneously, we focus on taking a general language model and specializing it on a particular dataset. Furthermore, the "context" vectors serve more to maintain local cohesion in the text than to influence style or content. In contrast, our work is focused on morphing a general language model to accommodate stylistic and topical aspects of small, focused datasets.

Other related efforts focus on adding parameters to general language models to capture certain topical or stylistic aspects of text. After training a general NNLM on 1.5 million beer reviews, Lipton et al. [20] freeze the learned weights in their

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

NNLM but train additional weights for additional inputs (e.g., the review's star rating and the style of the beer). Note that Lipton et al. must explicitly model and construct vectors for their auxiliary input. Their approach works well if one has in mind a specific use case which can be easily quantified, but this approach does not immediately adapt to scenarios where one wishes to specialize a general language model to a language model tailored to a particular style or topic.

There is also a collection of research which biases a general NNLM using a combination of variational autoencoders and generative adversarial networks [21]–[23]. Architecturally, each of these models resembles a standard NNLM with a variational autoencoder plugged in to bias the model toward a certain subset of the data. In contrast, we present a procedure which can bias a general NNLM without explicitly modeling or controlling latent variables (e.g., review score). Indeed, our goal is to train an NNLM in settings where only a small training set is available, whereas others wish to incorporate known attributes of their data into an NNLM.

Furthermore, the nature of the bias is more limited in most previous research than in this work. For example, much previous work on biasing NNLMs is focused on product reviews [20], [22], [23]. The biasing datasets used herein are much more specialized in relation to our general datasets than the set of positive reviews is to the set of all reviews.

The research described herein is most similar to concurrent work by Howard and Ruder [24]. However, unlike [24], which focuses on specific classification problems, we focus on general language modeling.

III. Approach

NNLMs transform input terms via a mapping from terms to an associated term-space vector representation. The vector representation corresponding to each successive input term is used to incrementally update a hidden state space vector representation of the sequence of terms seen by the network. At each step, this hidden state is used to derive a vector of probabilities representing the network's guess at the subsequent term in the sequence. When training a NNLM, we aim to minimize a loss function calculated from this vector of probabilities and the actual next term.

Our research is motivated by the insight that NNLM's word space transformations represent a mapping from the source language into and out of a "narrative space" representation, and that these mappings are likely language specific but domain independent. The fact that most of a NNLM's parameters are concentrated in these input and output layers (the mapping layers) implies that it may be beneficial to train all parameters of a NNLM on a large dataset and subsequently retrain only the interior layers of the network on our biasing datasets. Thus, the parameters responsible for the meanings of words are learned entirely on a large corpus while the (smaller number of) parameters responsible for the flow of language are learned on a large corpus but refined on small, specialized corpus characterized by a specific topic and/or style.

We separate our approach into three portions: preprocessing, training a general NNLM, and biasing the NNLM. Training neural networks requires one to fine-tune a myriad of hyperparameters (e.g., vector dimensions, learning rate, regularization parameters), and we have experimented with multiple values for each of these hyperparameters. In the discussion below, we mention only the values for the hyperparameters which were optimal for our language modeling tasks. All networks are trained using TensorFlow [25].

A. Preprocessing

During preprocessing, we produce training data in a format largely consistent with the term token structure of the Wall Street Journal portion of the Penn Treebank (PTB) corpus [26].

We begin by tokenizing the text using the PTBTokenizer from Stanford CoreNLP [27] with the settings "normalizeParentheses=False", "normalizeOtherBrackets=False", and "splitHyphenated=True". In addition to tokenization, this tokenizer splits text into its constituent sentences. We remove any "sentences" longer than 128 tokens; we found most of these to be either run-on or improper sentences (e.g., bulleted lists, bibliographic information).

We further convert all text to lowercase, and we add "<bos>" and "<eos>" tags to the beginning and end of each sentence. We train our models using a vocabulary of the 65,536 most commonly used terms. All terms not in our vocabulary are replaced with an "<unk>" token.

B. Training the general NNLM

We train two-layer LSTM NNLMs [1], [2] with peephole connections [28] on our general datasets. Furthermore, unless otherwise noted, all neural networks have input, hidden, and output layers with 2,048 dimensions.

Recall that each step of a NNLM takes as input a term in a sentence and produces as output a probability distribution representing the network's guess at the subsequent term in the sentence. Because calculating a probability for each term at each step of the LSTM can be computationally expensive, we use a sampled softmax loss function [29]. Using this loss function, it is necessary to compute probability estimates only for the correct term and a small number of incorrect terms. As suggested by Ji et al. [30], we choose to sample 1,024 incorrect terms, and the incorrect terms are each chosen with probability proportional to $count(t)^{0.4}$, where $count(t)$ represents the number of times term t occurs in the training dataset.

We regularize using dropout [31], a regularization technique which ignores a set percentage of hidden states during training. Explicitly, we randomly set 25% of the inputs of each hidden layer to zero. We train our networks using the Adam optimizer [32] with a batch size of 256. We train in two phases with empirically chosen learning rates. We first set the learning rate of the Adam optimizer to 0.001 and train until convergence. We then decrease the learning rate to 0.0001 and continue training until convergence.

TABLE I
DATASET STATISTICS

Dataset	Type	Number of Words	Number of Sentences
ENWiki	Gen	1,597,148,670	65,548,135
ENWiki_Computer	Top	4,531,972	146,159
ENWiki_Math	Top	5,081,695	170,400
ENWiki_Movies	Top	1,763,730	63,594
ENWiki_Science	Top	17,774,526	572,702
ENWiki_Sport	Top	34,224,253	1,241,570
SEWiki	Sty	18,597,300	1,042,226
SEWiki_Science	S+T	179,457	8,212
SEWiki_Sport	S+T	390,283	19,768
MovieReviews	S+T	1,494,946	63,577
WSJ	S+T	1,199,206	48,288

C. Training the biased NNLM

To bias a general NNLM, we continue the training procedure as discussed in Section III-B with a biasing dataset instead of the general dataset. When training the NNLM on a biasing dataset, we train only the interior layers of the network; the weights learned by the NNLM for the input and output layers during the first phase of training are fixed when biasing the NNLM.[1] With our chosen hyperparameters, our NNLMs each have 336 million parameters, of which only 67 million are dedicated to the internal transitions of the language model. Although the parameters controlling internal transitions represent a mere one-fifth of the original model, our networks are still prone to overfitting on our biasing datasets due to their small size. We thus regularly monitor the perplexity attained by the NNLM on the validation portion of our biasing data, and we stop the training procedure when this perplexity reaches a local minimum.

For a language model M and a language dataset D containing terms t, we measure the *perplexity* of M evaluated on D via:

$$\left(\prod_{t \in D} p(t\ ;\ M)\right)^{-1/|D|} = \exp\left[-\frac{1}{|D|}\sum_{t \in D}\log(p(t\ ;\ M))\right]$$

where $|D|$ represents the number of terms in D, and $p(t\ ;\ M)$ denotes the probability M assigns to the correct term t. Perplexity represents the inverse of the geometric mean of the probabilities assigned to the correct terms in the testing dataset. As such, lower perplexities imply better models.

IV. DATA

All considered datasets are listed in Table I. Our general dataset is the text of English Wikipedia (*ENWiki*).

We create biasing datasets of various forms. First, we construct subsets of *ENWiki* so that the sentences in each subset share some topical or stylistic similarity. To construct these datasets, all sentences containing certain unigrams or bigrams are considered to belong to a topically oriented biasing dataset. For example, we create a "math" subset of English Wikipedia, notated *ENWiki_Math* by collecting all sentences containing the terms "math", "mathematics", "algebra", "geometry", etc.

To study the relationship between the size of the biasing dataset and final performance of the model on its test set, we also construct extra subsets of various sizes of the training partition of *ENWiki_Science*, a collection of Wikipedia articles with a science focus.

To create a somewhat stylistically dissimilar dataset, we curate the text of Simple English Wikipedia (*SEWiki*). Simple English is a subset of English created to facilitate communication among those who are not fully fluent in Standard English. We further create topically oriented subsets of *SEWiki*.

We also explore datasets which have low stylistic similarity with *ENWiki*. These include *WSJ*, the Wall Street Journal portion of the Penn Treebank [26] (which has its own traditional training/validation/testing partitions), and *MovieReviews*, a set of movie reviews curated by Pang and Lee [33].

We split our datasets into a "training" partition to train the NNLMs, a "validation" partition to monitor model perplexity during training, and a "testing" partition which is used only to measure model performance as reported below. We use 80% of the data for the training partition, 10% of the data for the validation partition, and 10% of the data for the testing partition. For the topically biased subsets of a general dataset (e.g., *ENWiki_Science*), all partitions are subsets of the partitions of the corresponding general dataset.

Basic statistics for all of these datasets are listed in Table I. The "Type" column in Table I distinguishes our general datasets and those biasing datasets which we felt were by nature topically focused, stylistically distinct, or both. We emphasize that our biasing datasets are a fraction of the size of most corpora used to train NNLMs.

V. RESULTS

We first train three general NNLMs independently on *ENWiki*. We then bias each of these on each of our biasing datasets. In addition to calculating the perplexity of each of these NNLMs on each of the datasets, we also create an ensemble of the three general or biased NNLMs trained on the same dataset by averaging the predicted probability distributions at each step. Because we observed an improvement of roughly 10% when combining the three models trained using the same training procedure, we report results only for our ensemble models.

Herein we describe our results on the dataset discussed above. We also tested our algorithms against a large Twitter dataset and small, topically biased subsets. We omit discussion of the results on the Twitter corpora due to space constraints and because we obtained results similar to those shown below.

A. Perplexities between models

The perplexities attained by evaluating our models on each of our datasets are provided in Table II. Some datasets are inherently more complex and "perplexing" than others; to account for this disparity and to provide a fair comparison of models and datasets, we divide the values in each column by

[1] Fixing some parameter layers and training others is a feature used in image processing, and was already supported by Tensorflow.

TABLE II
PERPLEXITY ATTAINED BY EVALUATING STANDARD ENGLISH MODELS ON STANDARD ENGLISH DATASETS

Model \ Dataset	*ENWiki*	*EN_Comp*	*EN_Math*	*EN_Sci*	*EN_Sport*	*EN_Movies*	*MovReviews*	*SEWiki*	*SE_Sci*	*SE_Sport*
ENWiki	40.8	54.5	34.8	37.0	26.3	48.3	89.4	35.5	36.2	17.9
ENWiki_Computer	50.1	39.2	38.9	41.1	31.4	56.3	110.6	46.5	41.5	22.6
ENWiki_Math	51.8	60.9	24.3	39.7	32.1	63.4	116.8	47.9	39.6	23.8
ENWiki_Science	57.7	68.9	41.4	28.3	38.9	66.1	121.4	52.6	23.9	30.0
ENWiki_Sport	61.6	78.0	59.4	53.2	19.1	67.4	130.6	57.1	58.1	12.8
ENWiki_Movies	46.1	59.5	39.9	41.7	29.4	37.8	85.2	40.8	42.5	20.8
MovieReviews	54.5	73.2	47.1	48.6	36.1	53.1	59.4	47.9	49.2	27.1
SEWiki	46.5	63.2	43.2	42.3	30.8	51.6	99.4	29.8	31.2	13.9
SEWiki_Science	45.7	58.9	35.4	35.4	28.8	53.5	100.0	37.6	26.6	18.7
SEWiki_Sport	49.7	66.9	41.3	43.7	26.0	56.7	112.8	40.9	40.7	11.5

TABLE III
PERPLEXITY, NORMALIZED BY COLUMN, ATTAINED BY EVALUATING STANDARD ENGLISH MODELS ON STANDARD ENGLISH DATASETS

Model \ Dataset	*ENWiki*	*EN_Comp*	*EN_Math*	*EN_Sci*	*EN_Sport*	*EN_Movies*	*MovReviews*	*SEWiki*	*SE_Sci*	*SE_Sport*
ENWiki	1	1.39	1.43	1.31	1.38	1.28	1.50	1.19	1.36	1.56
ENWiki_Computer	1.23	1	1.60	1.45	1.64	1.49	1.86	1.56	1.56	1.97
ENWiki_Math	1.27	1.55	1	1.40	1.68	1.68	1.96	1.60	1.49	2.07
ENWiki_Science	1.41	1.76	1.70	1	2.04	1.75	2.04	1.76	0.90	2.61
ENWiki_Sport	1.51	1.99	2.44	1.88	1	1.78	2.20	1.91	2.18	1.11
ENWiki_Movies	1.13	1.52	1.64	1.47	1.54	1	1.43	1.37	1.60	1.81
MovieReviews	1.34	1.87	1.94	1.72	1.89	1.40	1	1.60	1.85	2.36
SEWiki	1.14	1.61	1.78	1.50	1.61	1.36	1.67	1	1.17	1.21
SEWiki_Science	1.12	1.50	1.45	1.25	1.51	1.41	1.68	1.26	1	1.63
SEWiki_Sport	1.22	1.71	1.70	1.54	1.36	1.50	1.90	1.37	1.53	1

the perplexity listed in the diagonal entry (i.e., the perplexity attained by evaluating the model on the testing partition of the model's training dataset); the result is shown in Table III.

We note that, with the exception of the *SEWiki_Science* model and dataset, the diagonal entries of Table III are the lowest values in their respective row and column. That is, all models but one can identify their own domain-specific data. We find it plausible that this exception is a result of the small yet relatively simple nature of the *SEWiki_Science* dataset. The *ENWiki_Science* model finds the sentences in the *SEWiki_Science* dataset relatively unperplexing given the somewhat more complex language but similar topic of the *ENWIKI_Science* dataset, and the small size of the *SEWiki_Science* dataset prevents the *SEWiki_Science* NNLM from outperforming the *ENWiki_Science* model.

Focusing on Table III and ignoring the general *ENWiki* and *SEWiki* datasets, the *ENWiki_Sports* model attains its second lowest score on the *SEWiki_Sports* dataset, and the *ENWiki_Movies* dataset attains its second lowest score on the *MovieReviews* dataset. The *SEWiki* model attains its second and third lowest scores on the considered subsets of the *SEWiki* dataset. This suggests that these models might be usable for style and topic identification.

B. Perplexity within model

Our training procedure produces better language models than training on the biasing datasets alone. Most of the biasing datasets are simply too small for training an NNLM from scratch. For example, after experimenting with multiple NNLM architectures, hyperparameters, and training procedures, the best perplexity attained by a single NNLM trained and evaluated only on the *SEWiki* dataset was 56.0, much higher than the perplexity of 32.4 attained by a single LSTM NNLM trained first on the *ENWiki* dataset and further biased on the *SEWiki* dataset. Similarly, each individual NNLM biased on the *WSJ* corpus attains perplexities on the testing partition of *WSJ* around 57.3, and the ensemble of NNLMs attains a perplexity of 51.2. Both of these perplexities are noticeably lower than the perplexity of 66.4 reported in [19].

C. Training data size analysis

To examine the effect of the size of the biasing dataset on model perplexity, we train models using subsets of various sizes of the training partition of *ENWiki_Science*. All models are tested using the full testing partition of *ENWiki_Science*. Results of this experiment are illustrated by Figure 1. The relationship between the size of the biasing dataset and perplexity is well captured by the power law

$$\text{perplexity} = 48.9 \times \text{numSentences}^{-0.040}.$$

D. Perplexity distributions

In any corpus, sentences may range from the canonical to the completely bizarre. When evaluated on the individual sentences from such a corpus, a NNLM will produce a variety of perplexity scores. We are interested in the distribution of these scores for different combinations of models and corpora. Thus, we use each model to calculate the perplexities of each

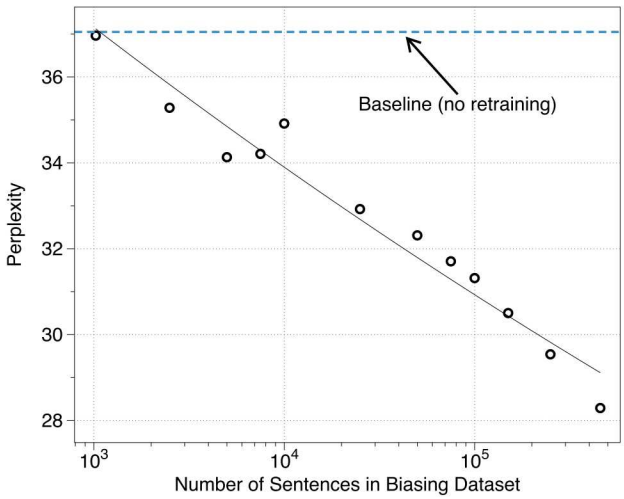

Fig. 1. Test perplexity on the testing partition of *ENWiki_Science* for models biased on subsets of the training partition of *ENWiki_Science* of various sizes.

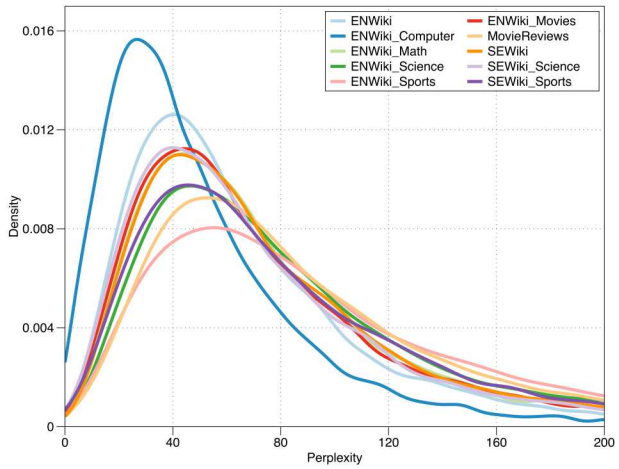

Fig. 2. Test sentence perplexity density estimates for the *ENWiki_Computer* dataset, as measured by each model.

sentence in each dataset. We use the collection of perplexity values for a given dataset and model to calculate a kernel density estimate (KDE) using a Gaussian smoothing kernel.

Figure 2 shows all models' KDEs calculated from the perplexity scores of the sentences from the *ENWiki_Computer* test dataset. This result is fairly typical across datasets; the matching model generally deems the data to have a lower, tighter distribution, whereas the non-matching models generally have similar and overlapping but generally higher-perplexity distributions.

E. Example outputs

Finally, for the amusement of the reader, we include an unbiased random sample of sentences generated by the *MovieReviews* model:

> Is there a mission to be able to save our environment?
>
> There are both men and women and rule both sexes, while each member of the class is given eight <unk> once lost in an environment from <unk> to trolls or aliens.
>
> However, Martin Scorsese's disappointing "the lantern of fools" bit proves the regular rotation can really be a bit more daunting.
>
> The wonder-riddled body guns and safe thrills are such satisfying crowd names...
>
> Edward island has featured a automated software storage system.

For contrast, we include the following sample of sentences from the *ENWiki_Math* model:

> In algebraic geometry, a space product, also known as a open set, is a topological space a for every subdivision of space that is compact, which is generally considered the regular topology.
>
> <unk> and <unk> both argue that change can be objectively better than self-improvement (affecting content on the cognition of perceptual thinking), which explains conceptual actions to atypical applications in computational statistics.
>
> We can view and explain changes in the vanishing bodies that are apparent in transcendental (or finite-time) geometry.
>
> Graduates' successful range of technology can be generated using various experiments, such as data processing, equipment, conservation strategies, statistics, intellectual output and book-keeping.
>
> It is also an important source of data for bounds on asymptotic properties of polynomial curves of probability.

VI. Conclusions and Future Work

In this work, we have introduced a technique to bias a neural network language model on a dataset which is a fraction of the size of the datasets traditionally used to train language models. Our technique can quickly transform an NNLM trained on a large dataset to an NNLM catered to capture the topical and stylistic facets of a small dataset.

We recognize that there have been many recent developments in neural network language modeling that are not reflected in this paper, and we look forward to applying our technique for biasing neural network language models to architectures other than LSTM models. Subsequent work will allow us to demonstrate the benefits of transfer learning in many common NLP applications. For example, our biased models could serve as one of many word sequence likelihood evaluators for various tasks.

We are interested in evaluating how confident topic classification of short documents should be in this framework based on a small set of sentence perplexities under competing hypotheses. Across each pair of the models and datasets tested, the estimated perplexity distribution seems to be well approximated by a gamma distribution. Further examination of this

observation will require formalizing the relationship between a NNLM's learned term probability distribution estimates and a gamma distribution fit to the resulting samples of sentence perplexities. This suggests a few potential approaches to text classification using model selection or empirical goodness-of-fit tests.

We believe our model could be combined with other approaches to address the long-standing problem of developing language models to understand humor, sarcasm, and idiomatic expressions. These nuances may be insoluble in a general language model, but with narrow, domain-focused models, it may eventually be possible to pick up on niche-specific cues revealing which statements are understood to be absurd when taken literally.

ACKNOWLEDGMENTS

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

REFERENCES

[1] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *arXiv preprint arXiv:1602.02410*, 2016.

[2] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," in *ICLR*, 2015.

[3] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP*, 2015, pp. 1412–1421.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in NIPS*, 2017, pp. 6000–6010.

[5] S. R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning, and C. Potts, "A fast unified model for parsing and sentence understanding," in *Transactions of the ACL*, vol. 1, 2016, pp. 1466–1477.

[6] C. Dyer, A. Kuncoro, M. Ballesteros, and N. A. Smith, "Recurrent neural network grammars," in *NAACL-HLT*, 2016, pp. 199–209.

[7] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan, "A persona-based neural conversation model," in *Transactions of the ACL*, 2016, pp. 994–1003.

[8] C. Xiong, S. Merity, and R. Socher, "Dynamic memory networks for visual and textual question answering," in *ICML*, 2016, pp. 2397–2406.

[9] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[10] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[12] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proceedings of EMNLP*, 2015, pp. 379–389.

[11] K. Filippova, E. Alfonseca, C. A. Colmenares, L. Kaiser, and O. Vinyals, "Sentence compression by deletion with LSTMs," in *ENMLP*, 2015, pp. 360–368.

[13] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, and L. Heck, "Contextual LSTM (CLSTM) models for large scale NLP tasks," *arXiv preprint arXiv:1602.06291*, 2016.

[14] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *NAACL-HLT*, 2016, pp. 1480–1489.

[15] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," *arXiv preprint arXiv:1312.3005*, 2013.

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[17] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1717–1724.

[18] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *EMNLP*, 2006, pp. 120–128.

[19] Y. Ji, T. Cohn, L. Kong, C. Dyer, and J. Eisenstein, "Document context language models," *arXiv preprint arXiv:1511.03962*, 2015.

[20] Z. C. Lipton, S. Vikram, and J. McAuley, "Capturing meaning in product reviews with character-level generative text models," in *CoRR*, 2015.

[21] J. Mueller, D. Gifford, and T. Jaakkola, "Sequence to better sequence: Continuous revision of combinatorial structures," in *ICML*, 2017, pp. 2536–2544.

[22] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola, "Style transfer from non-parallel text by cross-alignment," in *Advances in NIPS*, 2017, pp. 6833–6844.

[23] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," in *ICML*, 2017, pp. 1587–1596.

[24] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "TensorFlow: A system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.

[26] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of English: The Penn treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[27] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60.

[28] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, vol. 3, 2000, pp. 189–194.

[29] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," in *ACL-ICJNLP*, 2015.

[30] S. Ji, S. Vishwanathan, N. Satish, M. J. Anderson, and P. Dubey, "Blackout: Speeding up recurrent neural network language models with very large vocabularies," in *ICLR*, 2016.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[33] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the ACL*, 2004.