Self-Governing Neural Networks for On-Device Short Text Classification

Sujith Ravi
Google Research
Mountain View, CA, USA
sravi@google.com

Zornitsa Kozareva
Google
Mountain View, CA, USA
zornitsa@kozareva.com

Abstract

Deep neural networks reach state-of-the-art performance for wide range of natural language processing, computer vision and speech applications. Yet, one of the biggest challenges is running these complex networks on devices such as mobile phones or smart watches with tiny memory footprint and low computational capacity. We propose on-device Self-Governing Neural Networks (SGNNs), which learn compact projection vectors with local sensitive hashing. The key advantage of SGNNs over existing work is that they surmount the need for pre-trained word embeddings and complex networks with huge parameters. We conduct extensive evaluation on dialog act classification and show significant improvement over state-of-the-art results. Our findings show that SGNNs are effective at capturing low-dimensional semantic text representations, while maintaining high accuracy.

1 Introduction

Deep neural networks are one of the most successful machine learning methods outperforming many state-of-the-art machine learning methods in natural language processing (Sutskever et al., 2014), speech (Hinton et al., 2012) and visual recognition tasks (Krizhevsky et al., 2012). The availability of high performance computing has enabled research in deep learning to focus largely on the development of deeper and more complex network architectures for improved accuracy. However, the increased complexity of the deep neural networks has become one of the biggest obstacles to deploy deep neural networks on-device such as mobile phones, smart watches and IoT (Iandola et al., 2016). The main challenges with developing and deploying deep neural network models on-device are (1) the tiny memory footprint, (2) inference latency and (3) significantly low computational capacity compared to high performance computing systems such as CPUs, GPUs and TPUs on the cloud.

There are multiple strategies to build lightweight text classification models for on-device. One can create a small dictionary of common *input* → *category* mapping on the device and use a naive look-up at inference time. However, such an approach does not scale to complex natural language tasks involving rich vocabularies and wide language variability. Another strategy is to employ fast sampling techniques (Ahmed et al., 2012; Ravi, 2013) or incorporate deep learning models with graph learning like (Bui et al., 2017, 2018), which result in large models but have proven to be extremely powerful for complex language understanding tasks like response completion (Pang and Ravi, 2012) and Smart Reply (Kannan et al., 2016).

In this paper, we propose Self-Governing Neural Networks (SGNNs) inspired by projection networks (Ravi, 2017). SGNNs are on-device deep learning models learned via embedding-free projection operations. We employ a modified version of the locality sensitive hashing (LSH) to reduce input dimension from millions of unique words/features to a short, fixed-length sequence of bits. This allows us to compute a projection for an incoming text very fast, on-the-fly, with a small memory footprint on the device since we do not need to store the incoming text and word embeddings. We evaluate the performance of our SGNNs on Dialogue Act classification, because (1) it is an important step towards dialog interpretation and conversational analysis aiming to understand the intent of the speaker at every utterance of the conversation and (2) deep learning methods reached state-of-the-art (Lee and Dernoncourt, 2016; Khanpour et al., 2016; Tran et al., 2017; Ortega and Vu, 2017).

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 887–893
Brussels, Belgium, October 31 - November 4, 2018.

The main contributions of the paper are:

- Novel Self-Governing Neural Networks (SGNNs) for on-device deep learning for short text classification.
- Compression technique that effectively captures low-dimensional semantic text representation and produces compact models that save on storage and computational cost.
- On the fly computation of projection vectors that eliminate the need for large pre-trained word embeddings or vocabulary pruning.
- Exhaustive experimental evaluation on dialog act datasets, outperforming state-of-the-art deep CNN (Lee and Dernoncourt, 2016) and RNN variants (Khanpour et al., 2016; Ortega and Vu, 2017).

2 Self-Governing Neural Networks

We model the Self-Governing network using a *projection* model architecture (Ravi, 2017). The *projection* model is a simple network with dynamically-computed layers that encodes a set of efficient-to-compute operations which can be performed directly on device for inference. The model defines a set of efficient "projection" functions $\mathbb{P}(\vec{x}_i)$ that project each input instance $\vec{x}_i$ to a different space $\Omega_{\mathbb{P}}$ and then performs learning in this space to map it to corresponding outputs y_i^p. A very simple projection model comprises just few operations where the inputs $\vec{x}_i$ are transformed using a series of T projection functions $\mathbb{P}^1, ..., \mathbb{P}^T$ followed by a single layer of activations.

2.1 Model Architecture

In this work, we design a Self-Governing Neural Network (SGNN) using multi-layered locality-sensitive projection model. Figure 1 shows the model architecture of the on-device SGNN network. The *self-governing* property of this network stems from its ability to learn a model (e.g., a classifier) without having to initialize, load or store any feature or vocabulary weight matrices. In this sense, our method is a truly **embedding-free** approach unlike majority of the widely-used state-of-the-art deep learning techniques in NLP whose performance depends on embeddings pre-trained on large corpora. Instead, we use the projection functions to dynamically transform each input to a low-dimensional representation. Furthermore, we stack this with additional layers and non-linear activations to achieve deep, non-linear combinations of projections that permit the network to learn complex mappings from inputs x_i to outputs y_i. An SGNN network is shown below:

$$i_p = [\mathbb{P}^1(x_i), ..., \mathbb{P}^T(x_i)] \quad (1)$$

$$h_p = \sigma(W_p \cdot i_p + b_p) \quad (2)$$

$$h_t = \sigma(W_t \cdot h_{t-1} + b_t) \quad (3)$$

$$y_i = \texttt{softmax}(W_o \cdot h_k + b_o) \quad (4)$$

where, i_p refers to the output of projection operation applied to input x_i, h_p is applied to projection output, h_t is applied at intermediate layers of the network with depth k followed by a final `softmax` activation layer at the top. In a k-layer SGNN, h_t, where $t = p, p+1, ..., p+k-1$ refers to the k subsequent layers after the projection layer. W_p, W_t, W_o and b_p, b_t, b_o represent trainable weights and biases respectively.

The projection transformations use precomputed parameterized functions, i.e., they are not trained during the learning process, and their outputs are concatenated to form the hidden units for subsequent operations. Each input text x_i is converted to an intermediate feature vector (via raw text features such as skip-grams) followed by projections.

$$x_i \xrightarrow{\mathbb{F}} \vec{x_i} \xrightarrow{\mathbb{P}} [\mathbb{P}^1(x_i), ..., \mathbb{P}^T(x_i)] \quad (5)$$

On-the-fly Computation. The transformation step $\mathbb{F}$ dynamically extracts features from the raw input. Text features (e.g., skip-grams) are converted into feature-ids f_j (via hashing) to generate a sparse feature representation $\vec{x_i}$ of feature-id, weight pairs (f_j, w_j) . This intermediate feature representation is passed through projection functions $\mathbb{P}$ to construct projection layer i_p in SGNN. For this last step, a projection vector $\mathbb{P}^k$ is first constructed on-the-fly using a hash function with feature ids f_j in $\vec{x_i}$ and fixed seed as input, then dot product of the two vectors $< \vec{x_i}, \mathbb{P}^k >$ is computed and transformed into binary representation $\mathbb{P}^k(\vec{x_i})$ using $sgn(.)$ of the dot product.

As shown in Figure 1, both $\mathbb{F}$ and $\mathbb{P}$ steps are computed on-the-fly, i.e., no word-embedding or vocabulary/feature matrices need to be stored and looked up during training or inference. Instead feature-ids and projection vectors are dynamically computed via hash functions. For intermediate feature weights w_j, we use observed counts in

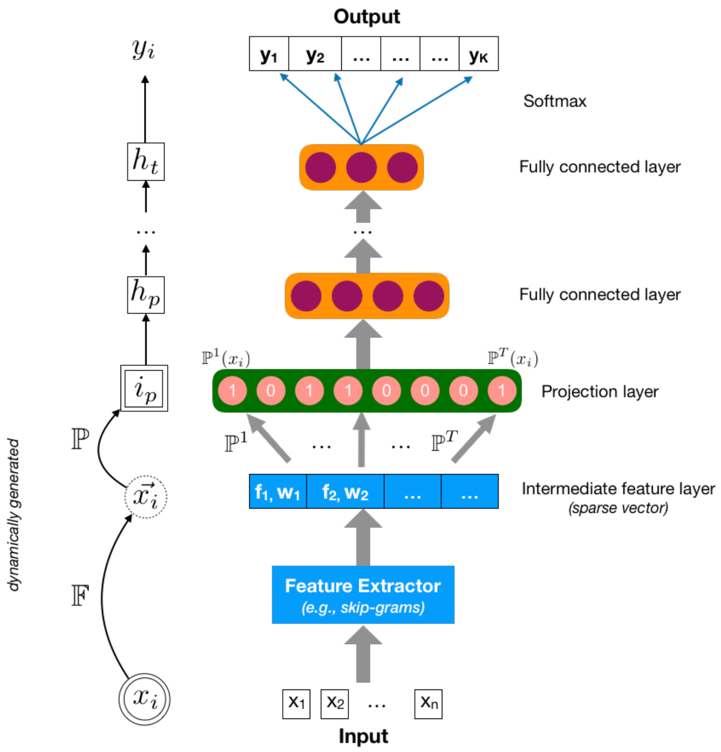

Figure 1: Self-Governing Neural Network (SGNN) architecture.

each input text and do not use pre-computed statistics like idf. Hence the method is embedding-free.

Model Optimization. The SGNN network is trained from scratch on the task data using a supervised loss defined wrt ground truth $\hat{y}_i$:

$$\mathcal{L}(.) = \sum_{i \in N} \texttt{cross} - \texttt{entropy}(y_i, \hat{y}_i) \quad (6)$$

During training, the network learns to choose and apply specific projection operations $\mathbb{P}^j$ (via activations) that are more predictive for a given task. The choice of the type of projection matrix $\mathbb{P}$ as well as representation of the projected space $\Omega_{\mathbb{P}}$ has a direct effect on computation cost and model size. We leverage an efficient randomized projection method and use a binary representation $\{0,1\}^d$ for $\Omega_{\mathbb{P}}$. This yields a drastically lower memory footprint both in terms of number and size of parameters.

Computing Projections. We employ an efficient randomized projection method for the projection step. We use locality sensitive hashing (LSH) (Charikar, 2002) to model the underlying projection operations in SGNN. LSH is typically used as a dimensionality reduction technique for clustering (Manning et al., 2008). LSH allows us to project similar inputs $\vec{x}_i$ or intermediate network layers into hidden unit vectors that are nearby in metric space. We use repeated binary hashing for $\mathbb{P}$ and apply the projection vectors to transform the input $\vec{x}_i$ to a binary hash representation denoted by $\mathbb{P}_k(\vec{x}_i) \in \{0,1\}$, where $[\mathbb{P}_k(\vec{x}_i)] := sgn[\langle \vec{x}_i, \mathbb{P}_k \rangle]$. This results in a d-bit vector representation, one bit corresponding to each projection row $\mathbb{P}_{k=1...d}$.

The same projection matrix $\mathbb{P}$ is used for training and inference. We never need to explicitly store the random projection vector $\mathbb{P}_k$ since we can compute them on the fly using hash functions over feature indices with a fixed row seed rather than invoking a random number generator. This also permits us to perform projection operations that are linear in the *observed* feature size rather than the *overall* feature or vocabulary size which can be prohibitively large for high-dimensional data, thereby saving both memory and computation cost. Thus, SGNN can efficiently model high-dimensional sparse inputs and large vocabulary sizes common for text applications instead of relying on feature pruning or other pre-processing heuristics employed to restrict input sizes in standard neural networks for feasible training. The binary representation is significant since this results in a significantly compact representation for the *projection* network parameters that in turn considerably reduces the model size.

SGNN Parameters. In practice, we employ T different projection functions $\mathbb{P}^{j=1...T}$, each resulting in d-bit vector that is concatenated to form the projected vector i_p in Equation 5. T and d vary depending on the *projection* network parameter configuration specified for $\mathbb{P}$ and can be tuned to trade-off between prediction quality and model size. Note that the choice of whether to use a single projection matrix of size $T \cdot d$ or T separate matrices of d columns depends on the type of projection employed (dense or sparse). For the intermediate feature step $\mathbb{F}$ in Equation 5, we use skip-gram features (3-grams with skip-size=2) extracted from raw text.

2.2 Training and Inference

We use the compact bit units to represent the *projection* in SGNN. During training, the network learns to move the gradients for points that are nearby to each other in the projected bit space $\Omega_{\mathbb{P}}$ in the same direction. SGNN network is trained end-to-end using backpropagation. Training can progress efficiently with stochastic gradient descent with distributed computing on high-performance CPUs or GPUs.

Complexity. The overall complexity for SGNN inference, governed by the projection layer, is $O(n \cdot T \cdot d)$, where n is the observed feature size (*not* overall vocabulary size) which is linear in input size, d is the number of LSH bits specified for each projection vector $\mathbb{P}_k$, and T is the number of projection functions used in $\mathbb{P}$. The model size (in terms of number of parameters) and memory storage required for the *projection* inference step is $O(T \cdot d \cdot C)$, where C is the number of hidden units in h_p in the multi-layer projection network and typically smaller than $T \cdot d$.

3 Datasets and Experimental Setup

3.1 Data Description

We conduct our experimental evaluation on two dialog act benchmark datasets.

- **SWDA:** Switchboard Dialog Act Corpus (Godfrey et al., 1992; Jurafsky et al., 1997) is a popular open domain dialogs corpus between two speakers with 42 dialogs acts.
- **MRDA:** ICSI Meeting Recorder Dialog Act Corpus (Adam et al., 2003; Shriberg et al., 2004) is a dialog corpus of multiparty meetings with 5 tags of dialog acts.

Datasets	Class	Vocab.	Train	Validation	Test
SwDA	42	20K	193K	23K	5K
MRDA	5	12K	78K	16K	15K

Table 1: Dialog Act Datasets Statistics

Table 1 summarizes dataset statistics. We use the train, validation and test splits as defined in (Lee and Dernoncourt, 2016; Ortega and Vu, 2017).

3.2 Experimental Setup

We setup our experimental evaluation, as follows: given a classification task and a dataset, we generate an on-device model. The size of the model can be configured (by adjusting the projection matrix $\mathbb{P}$) to fit in the memory footprint of the device, i.e. a phone has more memory compared to a smart watch. For each classification task, we report $Accuracy$ on the test set.

3.3 Hyperparameter and Training

For both datasets we used the following: 2-layer SGNN ($\mathbb{P}_{T=80,d=14}$ × FullyConnected$_{256}$ × FullyConnected$_{256}$), mini-batch size of 100, dropout rate of 0.25, learning rate was initialized to 0.025 with cosine annealing decay (Loshchilov and Hutter, 2016). Unlike prior approaches (Lee and Dernoncourt, 2016; Ortega and Vu, 2017) that rely on pre-trained word embeddings, we learn the projection weights on the fly during training, i.e word embeddings (or vocabularies) do not need to be stored. Instead, features are computed on the fly and are dynamically compressed via the projection matrices into projection vectors. These values were chosen via a grid search on development sets, we do not perform any other dataset-specific tuning. Training is performed through stochastic gradient descent over shuffled mini-batches with Nesterov momentum optimizer (Sutskever et al., 2013), run for 1M steps.

4 Results

Tables 2 and 3 show results on the SwDA and MRDA dialog act datasets. Overall, our SGNN model consistently outperforms the baselines and prior state-of-the-art deep learning models.

4.1 Baselines

We compare our model against a majority class baseline and Naive Bayes classifier (Lee and Dernoncourt, 2016). Our model significantly outperforms both baselines by 12 to 35% absolute.

4.2 Comparison against State-of-art Methods

We also compare our performance against prior work using HMMs (Stolcke et al., 2000) and recent deep learning methods like CNN (Lee and Dernoncourt, 2016), RNN (Khanpour et al., 2016) and RNN with gated attention (Tran et al., 2017).

To the best of our knowledge, (Lee and Dernoncourt, 2016; Ortega and Vu, 2017; Tran et al., 2017) are the latest approaches in dialog act classification, which also reported on the same data splits. Therefore, we compare our research against these works. According to (Ortega and Vu, 2017), prior work by (Ji and Bilmes, 2006) achieved promising results on the MRDA dataset, but since the evaluation was conducted on a different data split, it is hard to compare them directly.

For both SwDA and MRDA datasets, our SGNNs obtains the best result of 83.1 and 86.7 accuracy outperforming prior state-of-the-art work. This is very impressive given that we work with very small memory footprint and we do not rely on pre-trained word embeddings. Our study also shows that the proposed method is very effective for such natural language tasks compared to more complex neural network architectures such as deep CNN (Lee and Dernoncourt, 2016) and RNN variants (Khanpour et al., 2016; Ortega and Vu, 2017). We believe that the compression techniques like locality sensitive projections jointly coupled with non-linear functions are effective at capturing low-dimensional semantic text representations that are useful for text classification applications.

4.3 Discussion on Model Size and Inference

LSTMs have millions of parameters, while our on-device architecture has just 300K parameters (order of magnitude lower). Most deep learning methods also use large vocabulary size of 10K or higher. Each word embedding is represented as 100-dimensional vector leading to a storage requirement of $10,000 \times 100$ parameter weights just in the first layer of the deep network. In contrast, SGNNs in all our experiments use a fixed 1120-dimensional vector regardless of the vocabulary or feature size, dynamic computation results in further speed up for high-dimensional feature spaces. This amounts to a huge savings in storage and computation cost wrt FLOPs (floating point operations per second).

Method	Acc.
Majority Class (baseline) (Ortega and Vu, 2017)	33.7
Naive Bayes (baseline) (Khanpour et al., 2016)	47.3
HMM (Stolcke et al., 2000)	71.0
DRLM-conditional training (Ji and Bilmes, 2006)	77.0
DRLM-joint training (Ji and Bilmes, 2006)	74.0
LSTM (Lee and Dernoncourt, 2016)	69.9
CNN (Lee and Dernoncourt, 2016)	73.1
Gated-Attention&HMM (Tran et al., 2017)	74.2
RNN+Attention (Ortega and Vu, 2017)	73.8
RNN (Khanpour et al., 2016)	80.1
SGNN: Self-Governing Neural Network (ours)	**83.1**

Table 2: SwDA Dataset Results

Method	Acc.
Majority Class (baseline)(Ortega and Vu, 2017)	59.1
Naive Bayes (baseline) (Khanpour et al., 2016)	74.6
Graphical Model (Ji and Bilmes, 2006)	81.3
CNN (Lee and Dernoncourt, 2016)	84.6
RNN+Attention(Ortega and Vu, 2017)	84.3
RNN (Khanpour et al., 2016)	**86.8**
SGNN: Self-Governing Neural Network (ours)	**86.7**

Table 3: MRDA Dataset Results

5 Conclusion

We proposed Self-Governing Neural Networks for on-device short text classification. Experiments on multiple dialog act datasets showed that our model outperforms state-of-the-art deep leaning methods (Lee and Dernoncourt, 2016; Khanpour et al., 2016; Ortega and Vu, 2017). We introduced a compression technique that effectively captures low-dimensional semantic representation and produces compact models that significantly save on storage and computational cost. Our approach does not rely on pre-trained embeddings and efficiently computes the projection vectors on the fly. In the future, we are interested in extending this approach to more natural language tasks. For instance, we built a multilingual SGNN model for customer feedback classification (Liu et al., 2017) and obtained 73% on Japanese, close to best performing system on the challenge (Plank, 2017). Unlike their method, we did not use any pre-processing, tagging, parsing, pre-trained embeddings or other resources.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback and suggestions.

References

Janin Adam, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The icsi meeting corpus. In *Proceedings of the 5TH SIGdial Workshop on Discourse and Dialogue*, pages 364–367.

Amr Ahmed, Sujith Ravi, Alex J. Smola, and Shravan M. Narayanamurthy. 2012. Fastex: Hash clustering with exponential families. In *Advances in Neural Information Processing Systems 25*, pages 2798–2806.

Thang D. Bui, Sujith Ravi, and Vivek Ramavajjala. 2017. Neural graph machines: Learning neural networks using graphs. *CoRR*, abs/1703.04818.

Thang D. Bui, Sujith Ravi, and Vivek Ramavajjala. 2018. Neural graph learning: Training neural networks using graphs. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 64–71.

Moses S. Charikar. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 380–388, New York, NY, USA. ACM.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'92, pages 517–520. IEEE Computer Society.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. 2012. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*.

Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360.

Gang Ji and Jeff Bilmes. 2006. Backoff model training using partially observed data: Application to dialog act tagging. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 280–287.

Daniel Jurafsky, Rebecca Bates, Rachel Martin Noah Coccaro, Marie Meteer, Klaus Ries, Elizabeth Shriberg, Audreas Stolcke, Paul Taylor, and Van Ess-Dykema. 1997. Automatic detection of discourse structure for speech recognition and understanding. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 88–95.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 955–964.

Hamed Khanpour, Nishitha Guntakandla, and Rodney Nielsen. 2016. Dialogue act classification in domain-independent conversations using a deep recurrent neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2012–2021.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–520.

Chao-Hong Liu, Yasufumi Moriya, Alberto Poncelas, and Declan Groves. 2017. Ijcnlp-2017 task 4: Customer feedback analysis. In *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 26–33, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Ilya Loshchilov and Frank Hutter. 2016. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Daniel Ortega and Ngoc Thang Vu. 2017. Neural-based context representation learning for dialog act classification. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 247–252.

Bo Pang and Sujith Ravi. 2012. Revisiting the predictability of language: Response completion in social media. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1489–1499.

Barbara Plank. 2017. All-in-1 at ijcnlp-2017 task 4: Short text classification with one model for all languages. In *Proceedings of the IJCNLP 2017, Shared*

Tasks, pages 143–148, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Sujith Ravi. 2013. Scalable decipherment for machine translation via hash sampling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 362–371. Association for Computational Linguistics.

Sujith Ravi. 2017. Projectionnet: Learning efficient on-device deep networks using neural projections. *CoRR*, abs/1708.00630.

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The icsi meeting recorder dialog act (mrda) corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 26(3):339–373.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pages III–1139–III–1147.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112.

Quan Hung Tran, Gholamreza Haffari, and Ingrid Zukerman. 2017. A generative attentional neural network model for dialogue act classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 524–529.