ProFormer: Towards On-Device LSH Projection Based Transformers

Chinnadhurai Sankar
Mila, Université de Montréal*
Montreal, QC, Canada
chinnadhurai@gmail.com

Sujith Ravi
Amazon Alexa†
Sunnyvale, CA, USA
sravi@sravi.org

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

Abstract

At the heart of text based neural models lay word representations, which are powerful but occupy a lot of memory making it challenging to deploy to devices with memory constraints such as mobile phones, watches and IoT. To surmount these challenges, we introduce ProFormer – a projection based transformer architecture that is faster and lighter making it suitable to deploy to memory constraint devices and preserve user privacy. We use LSH projection layer to dynamically generate word representations on-the-fly without embedding lookup tables leading to significant memory footprint reduction from $\mathcal{O}(V.d)$ to $\mathcal{O}(T)$, where V is the vocabulary size, d is the embedding dimension size and T is the dimension of the LSH projection representation. We also propose a *local projection attention (LPA)* layer, which uses self-attention to transform the input sequence of N LSH word projections into a sequence of N/K representations reducing the computations quadratically by $\mathcal{O}(K^2)$.

We evaluate ProFormer on multiple text classification tasks and observed improvements over prior state-of-the-art on-device approaches for short text classification and comparable performance for long text classification tasks. ProFormer is also competitive with other popular but highly resource-intensive approaches like BERT and even outperforms small-sized BERT variants with significant resource savings – reduces the embedding memory footprint from 92.16 MB to 1.7 KB and requires $16\times$ less computation overhead, which is very impressive making it the fastest and smallest on-device model.

1 Introduction

Transformers (Vaswani et al., 2017) based architectures like BERT (Devlin et al., 2018), XL-net (Yang et al., 2019), GPT-2 (Radford et al., 2019), MT-DNN (Liu et al., 2019a), RoBERTA (Liu et al., 2019b) reached state-of-the-art performance on tasks like machine translation (Arivazhagan et al., 2019), language modelling (Radford et al., 2019), text classification benchmarks like GLUE (Wang et al., 2018). However, these models require huge amount of memory and need high computational requirements making it hard to deploy to small memory constraint devices such as mobile phones, watches and IoT. Recently, there have been interests in making BERT lighter and faster (Sanh et al., 2019; McCarley, 2019). In parallel, recent on-device works like SGNN (Ravi and Kozareva, 2018), SGNN++ (Ravi and Kozareva, 2019) and (Sankar et al., 2019) produce lightweight models with extremely low memory footprint. They employ a modified form of LSH projection to dynamically generate a fixed binary projection representation, $\mathbb{P}(x) \in [0,1]^T$ for the input text x using word or character n-grams and skip-grams features, and a 2-layer MLP + softmax layer for classification. As shown in (Ravi and Kozareva, 2018) these models are suitable for short sentence lengths as they compute T bit LSH projection vector to represent the entire sentence. However, (Kozareva and Ravi, 2019) showed that such models cannot handle long text due to significant information loss in the projection operation.

On another side, recurrent architectures represent long sentences well, but the sequential nature of the computations increases latency requirements and makes it difficult to launch on-device. Recently, self-attention based architectures like BERT (Devlin et al., 2018) have demonstrated remarkable success in capturing long term dependencies in the input text via purely attention mechanisms. BERT's model architecture is a multi-layer bidirectional Transformer encoder based on the original implementation in (Vaswani et al., 2017). The

*Work done during internship at Google
†Work done while at Google AI

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2823–2828
April 19 - 23, 2021.

self-attention scores can be computed in parallel as they do not have recurrent mechanisms. But usually these architectures are very deep and the amount of computation is quadratic in the order of $\mathcal{O}(L \cdot N^2)$, where L is the number of layers (Transformer blocks) and N is the input sentence length. Straightforward solutions like reducing the number of layers is insufficient to launch transformers on-device due to the large memory and quadratic computation requirements.

In this paper, we introduce a projection-based neural architecture ProFormer that is designed to (a) be efficient and learn compact neural representations (b) handle out of vocabulary words and misspellings (c) drastically reduce embedding memory footprint from hundreds of megabytes to few kilobytes and (d) reduce the computation overhead *quadratically* by introducing a local attention layer which reduces the intermediate sequence length by a constant factor, K. We achieve this by bringing the best of both worlds by combining LSH projection based representations (for low memory footprint) and self-attention based architectures (to model dependencies in long sentences). To tackle computation overheard in the transformer based models, we reduce the number of self-attention layers and additionally introduce an intermediate local projection attention (LPA) to quadratically reduce the number of self-attention operations. The main contributions of our paper are:

- We propose novel on-device neural network called ProFormer which combines LSH projection based text representations, with transformer architecture and locally projected self-attention mechanism that captures long range sentence dependencies while yielding low memory footprint and low computation overhead.
- ProFormer reduces the computation overhead $\mathcal{O}(L \cdot N^2)$ and latency in multiple ways: by reducing the number of layers L from twelve to two and introducing new local projection attention layer that decreases number of self-attention operations by a quadratic factor.
- ProFormer is light weigh compact on-device model, while BERT on-device still needs huge embedding table (92.16 MB for $V = 30k$, $d = 768$) with number of computation flops in the order of $\mathcal{O}(L \cdot N^2)$, where L is the the number of layers, N is the number of words in the input sentence.
- We conduct empirical evaluations and comparisons against state-of-the-art on-device and prior deep learning approaches for short and long text classification. Our model ProFormer reached state-of-art performance for short text and comparable performance for long texts, while maintaining small memory footprint and computation requirements.

2 ProFormer: LSH Projection based Transformers

In this section, we show the overall architecture of ProFormer in Figure 1. ProFormer consists of multiple parts: (1) word-level Locality Sensitive Hashing (LSH) projection layer, (2) local projection attention (LPA) layer, (3) transformer layer (Devlin et al., 2018) and (4) a max-pooling + classifier layer. Next, we describe each layer in detail.

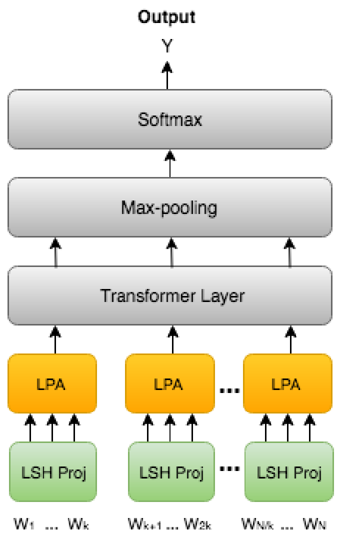

Figure 1: ProFormer: Our **Pro**jection Trans**former** Network Architecture

2.1 LSH Projection Layer

It is a common practice to represent each word in the input sentence, $\mathbf{x} = [w_1, w_2, \cdots, w_N]$ as an embedding vector based on its one-hot representation. Instead, we adopt LSH projection layer from (Ravi, 2017, 2019) which dynamically generates a T bit representation, $\mathbb{P}(w_i) \in [0, 1]^T$ for the input word, w_i based on its morphological features like n-grams, skip-grams from the current and context words, parts-of-speech tags, etc.

Since the LSH projection based approach does not rely on embedding lookup tables to compute word representation, we obtain significant memory savings of the order, $O(V \cdot d)$, where V is the vocabulary size and d is the embedding dimension. For instance, the embedding look-up table occupies 92.16 MB ($V = 30k$, $d = 768$ (Devlin et al., 2018)), while the LSH projection layer requires only $\approx$ 1.7 KB ($T = 420$) as shown in Table 1.

Models	Embedding memory	Computations
BERT	$\mathcal{O}(V.d)$	$\mathcal{O}(N^2)$
ProFormer (our model)	$\mathcal{O}(T)$	$\mathcal{O}(N^2/K^2)$

Table 1: Memory and computations overhead comparison between BERT (Devlin et al., 2018) and ProFormer (our model). N is the number of words in the input. For $V = 30k$, $d = 768$, $T = 420$, BERT's embedding table occupies 92.16 MB while ProFormer requires only **1.7** KB. For $K = 4$, we reduce the BERT computation overhead by **16** times.

2.2 Local Projection Attention (LPA) Layer

The LPA layer shown in Figure 2 consists of a single layer multi-headed self-attention layer similar to the Transformer architecture in (Vaswani et al., 2017) followed by a max-pooling layer yielding a compressed representation of K input words, $[w_1, w_2, \cdots w_K]$.

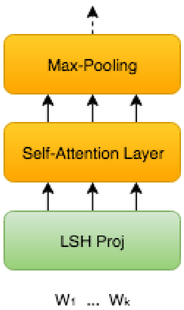

Figure 2: Local Projection Attention (LPA) layer.

The LPA layer transforms the N word-level projections, $\mathbb{P}(w_i)$ to a sequence of N/K representations as in Equation 1.

$$[\mathbb{P}(w_1), \cdots \mathbb{P}(w_N)]_N \longrightarrow [\mathcal{LPA}(\mathbb{P}(w_{1:K})), \cdots \mathcal{LPA}(\mathbb{P}(w_{N/K:N}))]_{N/K} \quad (1)$$

where $\mathcal{LPA}$ consists of the self-attention and max-pooling operation, K is a *Group factor*[1]. We equally divide the N word-level LSH projection representations into N/K groups of size K. The LPA layer compresses each group of K word representations into $\mathcal{LPA}(\mathbb{P}(w_{1:K})) \in \mathbb{R}^d$ yielding N/K representations in total. The LPA layer reduces the self-attention computation overhead in the subsequent transformer layer (Vaswani et al., 2017) by $\mathcal{O}(K^2)$.

[1]We choose K such that N is divisible by K.

2.3 Transformer Layer

This layer consists of 2-layer bidirectional Transformer encoder based on the original implementation described in (Vaswani et al., 2017). This layer transforms the N/K input representations from the LPA layer described in the previous sub-section into N/K output representations. In this layer, we reduce both the computation overhead and memory footprint by reducing the number of layers from L to 2 reducing the computation overhead by $\mathcal{O}(L/2)$ (6 times in the case of 12-layer *BERT-base* model).

2.4 Max-Pooling and Classification Layer

We summarize the N/K representations from the transformer layer to get a single d dimensional vector by max-pooling across the N/K time-steps, followed by a softmax layer to predict the output class Y.

3 Datasets & Experimental Setup

In this section, we describe our datasets and experimental setup. We use text classification datasets from state-of-the-art on-device evaluations such as: MRDA (Shriberg et al., 2004) and ATIS (Tür et al., 2010), AG News (Zhang et al., 2015a) and Yahoo! Answers (Zhang et al., 2015a). Table 2 shows the characteristics of each dataset.

Tasks	# Classes	Avg-len	Train	Test
MRDA (Dialog act)	6	8	78k	15k
ATIS (Intent prediction)	21	11	$4.4k$	$0.89k$
AG (News Categorization)	4	38	120k	7.6k
Y!A (Yahoo! Answers Categorization)	10	108	1400k	60k

Table 2: Classification Dataset Characteristics

We train ProFormer on multiple classification tasks individually and report *Accuracy* on corresponding test sets. We fix the projection size, $T = 420$, n-gram size=5, skip-gram size=1 for the LSH projection operation, $\mathbb{P}$. For the LPA layer, We experiment with two values for $K = 1, 4$, where $K = 1$ corresponds to the null operation in the LPA layer which just passes the word LSH projection representation to the Transformer layer. For the transformer layer, we fix the number of layers, $L = 2$ and set all layer sizes, $d = 768$ (including the intermediate size for the dense layer).[2]

[2]The rest of the parameters are same as the one used in

We compare our model with previous state of the art neural architectures, including on-device approaches. We also fine-tune the pretrained 12-layer *BERT-base* model (Devlin et al., 2018) on all classification tasks and compare to our model. *BERT-base* consists 12-layers of transformer blocks (Vaswani et al., 2017) and is pretrained in an unsupervised manner on a large corpus (BooksCorpus (Zhu et al., 2015) and English WikiPedia) using masked-language model objective. We fine-tune the pretrained *BERT-base* (Devlin et al., 2018) to each of the classification tasks. For training, we use Adam with learning rate of 1e-4, β_1=0.9, β_2=0.999, $L2$ weight decay of 0.01, learning rate warmup over the first $10,000$ steps, and linear decay of the learning rate. We use dropout probability of 0.1 on all layers and training batch size of 256. For further comparison, we also trained much smaller BERT baselines with 2-layers of transformer blocks and smaller input embedding sizes.

4 Results

Tables 3 and 4 show the results on the ATIS & MRDA short text classification and AG & Y!A long text classification tasks. We compare our approach, ProFormer against prior state-of-the-art on-device works, fine-tuned *BERT-base*, smaller 2-layer *BERT* variants and other non-on-device neural approaches.

Overall, our model ProFormer improved upon non-on-device neural models while keeping very small memory footprint and high accuracy. This is very impressive since ProFormer can be directly deployed to memory constraint devices like phones, watches and IoT while still maintaining high accuracy. ProFormer also improved upon prior on-device state-of-the-art neural approaches like SGNN (Ravi and Kozareva, 2018) and SGNN++ (Ravi and Kozareva, 2019) reaching over 35% improvement on long text classification. Similarly it improved over on-device ProSeqo (Kozareva and Ravi, 2019) models for all datasets and reached comparable performance on MRDA. In addition to the quality improvements, ProFormer also keeps smaller memory footprint than ProSeqo, SGNN and SGNN++.

In addition to the non-on-device and on-device neural comparisons, we also compare against *BERT-base* and other smaller variants. Our experiments show that ProFormer outperforms the small *BERT* baselines on all tasks. Moreover, although the 12-layer fine-tuned *BERT-base* (Devlin et al., 2018) model converged to the state-of-the-art in almost all of the tasks, ProFormer converges to $\approx 97.2\%$ *BERT-base*'s performance on an average while occupying only 13% of *BERT-base's* memory. ProFormer has 14.4 million parameters, while BERT-base has 110 million. For fair comparison, we also test ProFormer with $K = 4$, which only occupies 38.4% the memory footprint of 2-layer BERT-base model and reduces the computation overhead by 16 times. The embedding look up table occupies nearly 23 million parameters out of 38 million parameters in the 2-layer BERT model. We notice that K=4 model performs slightly worse than K=1 indicating information loss in the LPA layer. Overall, our experiments demonstrate that ProFormer reaches better performances that prior non-on-device and on-device neural approaches, and comparable performance to BERT-base models while preserving smaller memory footprint.

Models	MRDA	ATIS
ProFormer (K=1) (our model)	**89.3**	**98.2**
ProFormer (K=4) (our model)	86.7	97.0
BERT-base + fine-tuned (Devlin et al., 2018) *(12-layers, embedding size = 768)*	**90.1**	**98.3**
BERT *(2-layer, embedding size = 560)*	77.0	94.0
BERT *(2-layer, embedding size = 840)*	76.8	95.0
ProSeqo (Kozareva and Ravi, 2019)(on-device)	**90.1**	97.8
SGNN++ (Ravi and Kozareva, 2019)(on-device)	87.3	93.7
SGNN (Ravi and Kozareva, 2018)(on-device)	86.7	88.9
RNN(Khanpour et al., 2016)	86.8	-
RNN+Attention(Ortega and Vu, 2017)	84.3	-
CNN(Lee and Dernoncourt, 2016)	84.6	-
GatedIntentAtten.(Goo et al., 2018)	-	94.1
GatedFullAtten.(Goo et al., 2018)	-	93.6
JointBiLSTM(Hakkani-Tur et al., 2016)	-	92.6
Atten.RNN(Liu and Lane, 2016)	-	91.1

Table 3: Short text classification results.

Models	AG	Y!A
ProFormer (K=1) (our model)	**92.0**	**72.8**
ProFormer (K=4) (our model)	91.5	71.1
BERT-base + fine-tuned (Devlin et al., 2018) *(12-layers, embedding size = 768)*	**94.5**	**73.8**
BERT *(2-layer, embedding size = 560)*	82.3	-
BERT *(2-layer, embedding size = 840)*	83.3	-
ProSeqo (Kozareva and Ravi, 2019)(on-device)	91.5	72.4
SGNN (Ravi and Kozareva, 2018)(on-device)	57.6	36.5
FastText-full (Joulin et al., 2016)	**92.5**	72.3
CharCNNLargeWithThesau.(Zhang et al., 2015b)	90.6	71.2
CNN+NGM (Bui et al., 2018)	86.9	-
LSTM-full (Zhang et al., 2015b)	86.1	70.8

Table 4: Long text classification results.

5 Conclusion

We proposed a novel on-device neural network ProFormer, which combines LSH projection based

$bert\_config.json$ in *BERT-base* model (Devlin et al., 2018) text representations, with trans-former architecture and locally projected self-attention mechanism that captures long range sentence dependencies. Overall, ProFormer yields low memory footprint and reduces computations quadratically. In series of experimental evaluations on short and long text classifications we show that ProFormer improved upon prior neural models and on-device work like SGNN (Ravi and Kozareva, 2018), SGNN++ (Ravi and Kozareva, 2019) and ProSeqo (Kozareva and Ravi, 2019). ProFormer reached comparable performance to our BERT-base implementation, however it produced magnitudes more compact models than BERT-base. This is very impressive showing both effectiveness and compactness of our neural model.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757. Association for Computational Linguistics.

Dilek Hakkani-Tur, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Vivian Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association (INTERSPEECH 2016)*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651.

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.

Zornitsa Kozareva and Sujith Ravi. 2019. ProSeqo: Projection sequence networks for on-device text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3892–3901.

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.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *Proceedings of The 17th Annual Meeting of the International Speech Communication Association (INTERSPEECH 2016)*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. *CoRR*, abs/1901.11504.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

J. Scott McCarley. 2019. Pruning a bert-based question answering model. *ArXiv*, abs/1910.06360.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

Sujith Ravi. 2019. Efficient on-device models using neural projections. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5370–5379.

Sujith Ravi and Zornitsa Kozareva. 2018. Self-governing neural networks for on-device short text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 804–810.

Sujith Ravi and Zornitsa Kozareva. 2019. On-device structured and context partitioned projection networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3784–3793, Florence, Italy. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Chinnadhurai Sankar, Sujith Ravi, and Zornitsa Kozareva. 2019. Transferable neural projection representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3355–3360, Minneapolis, Minnesota. Association for Computational Linguistics.

Elizabeth Shriberg, Rajdip Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proceedings of the SIGDIAL 2004 Workshop, The 5th Annual Meeting of the Special Interest Group on Discourse and Dialogue, April 30 - May 1, 2004, Cambridge, Massachusetts, USA*, pages 97–100.

Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck. 2010. What is left to be understood in atis? In *Proceedings of 2010 IEEE Spoken Language Technology Workshop (SLT)*, pages 19–24.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015a. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, pages 649–657.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657. MIT Press.

Yukun Zhu, Ryan Kiros, Richard S. Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 19–27.