Graph Neural Networks for Adapting Off-the-shelf General Domain Language Models to Low-Resource Specialised Domains

Merieme Bouhandi[1], **Emmanuel Morin**[1] and **Thierry Hamon**[2]
[1]LS2N, UMR CNRS 6004, Nantes Université, Nantes, France
[2]LISN, Université Paris-Saclay & Université Sorbonne Paris Nord, France
{merieme.bouhandi, emmanuel.morin}@ls2n.fr
thierry.hamon@limsi.fr

Abstract

Language models encode linguistic proprieties and are used as input for more specific models. Using their word representations as-is for specialised and low-resource domains might be less efficient. Methods of adapting them exist, but these models often overlook global information about how words, terms, and concepts relate to each other in a corpus due to their strong reliance on attention. We consider that global information can influence the results of the downstream tasks, and combination with contextual information is performed using graph convolution networks or GCN built on vocabulary graphs. By outperforming baselines, we show that this architecture is profitable for domain-specific tasks.

1 Introduction

Numerous types of word vectors are used as word representations for NLP tasks. These vectors encode useful semantic proprieties and are often used as weights or input of generic task models. For quite some time, Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) were the go-to off-the-shelf embeddings that many systems used. FastText (Bojanowski et al., 2017) came as a way to deal with OOV words and still offers a better representation for words than most systems since it takes into account the morphological complexity of words by dealing with n-grams instead of whole words. Over the last few years, a new generation of deep neural approaches brought forth by transformers has brought significant improvements in many downstream applications. Language models, like BERT (Devlin et al., 2019), already encode so much knowledge and can capture semantic and syntactic information remarkably well (Coenen et al., 2019). They can be further trained on new tasks for adapting it to a new task or more specialised domains and further improve the quality of the representations (Peters et al., 2019).

Language models are trained over massive general domain corpora (Graff et al., 2003; Zhu et al., 2015) by optimising an objective that predicts the local contexts, captures linguistic units' distributional properties along the way, and address polysemy issues. Their quality can thus be arguably correlated to the volume of data available. However, in the case of specialised domains, the corpora are generally relatively modest in size, and these methods might be less efficient. It must be noted that if this claim is not always valid for all domains, especially for the English language, it is undoubtedly almost always true for other much less-resourced languages (Eisenschlos et al., 2019).

When using neural-based models, the conventional way of integrating further knowledge about specialised domains into models pre-trained on general corpora is to leverage pre-training by doing transfer learning and fine-tuning the model, tweaking its original weights to suit the tasks at hand better. If fine-tuning BERT is the most used adaptation method, it will rapidly hit a performance ceiling if the domain is too specialised or small, and some methods (Schick and Schütze, 2020) exist to tackle this problem. BERT will heavily rely on the context to build representations of too specialised or rare words. Each layer of the encoder's attention mechanism enriches the new representation of the input data with contextual information by paying attention to different parts of the text. Nonetheless, this particular feature of BERT may make it more challenging for it to consider the more global place a word occupies within a corpus's vocabulary, especially for these words.

Many data structures are hierarchical or graph structures in nature, such as social networks, paper citation networks, ontologies and semantic relations, such as hypernymy of hyponymy. Global relations between words within a sentence, a document or a corpus can be represented as a graph. Using such graphs as inputs, Graph Neural Net-

Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022), pages 36 - 42
July 15, 2022 ©2022 Association for Computational Linguistics

work (GNN) (Wu et al., 2021) captures general knowledge about the words and how they interact in a corpus. Several variants of GNN for text classification tasks exist. As presented by (Kipf and Welling, 2017), Graph Convolutional networks (GCN) is an approach for semi-supervised learning on graph-structured data based on an efficient variant of convolutional neural networks that operate directly on graphs. GCN is typically built using an adjacency matrix (based on a given relationship graph). The GCN's central idea is to take the weighted average of a node and all its neighbours during the convolution operation and uses both node features and the structure for the training. Text GCN (Yao et al., 2019) is a particular version of GCN, jointly learning both words and documents embeddings, and suited for situations with less training data. Graph methods do not encode positional information, and when information about position or context is needed, it might not be enough. Hence, pairing a GCN with a model that can grasp contextual information, like BERT, seems necessary.

There has not been much work trying to combine BERT and GNN. Since experts make inferences with relevant domain knowledge when performing domain-specific tasks, previous work has been conducted to integrate this knowledge into language models using knowledge graphs. Knowledge-enabled language representation model K-BERT (Liu et al., 2019) injects triples from a knowledge graph into the sentences as domain knowledge. BERT-MK (He et al., 2020) also takes into account knowledge graph contextualised knowledge. (Shang et al., 2019) embedded a medical ontology with Graph Attention Networks (GAT) and combined it with BERT for medication recommendation. (Jeong et al., 2020) concatenates the output of GCN and the output of BERT for citation recommendation tasks. Obviously, these methods presuppose the existence of a knowledge graph or an ontology. These resources are expensive to build and do not always exist for the domain and task at hand. Vocabulary graphs, on the other hand, are easy to build. (Lu et al., 2020) propose VGCN-BERT, a model which combines BERT with a Vocabulary GCN (VGCN), and where local and global information interacts from the first layer of BERT down, building an augmented representation jointly.

We build upon their work to adapt BERT to our specialised tasks and corpora. As (Lu et al., 2020) conveniently pointed out in their work, the utility of capturing global dependencies with a graph embedding instead of conventional non-contextualised embedding models (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) can be questioned (Srinivasan and Ribeiro, 2020). These methods provide additional information, but these models' small text window limits the connections between words. Long-range connections are more easily captured with GCN. In addition, by building a graph on a task-specific corpus, task-dependent dependencies are captured, in addition to the general dependencies already encoded in the pre-trained models.

2 Intrintic Evaluation Tasks

I2B2 (Uzuner et al., 2011) deals with automatic medical concept extraction and deals with the extraction of concepts (problems, tests and treatments) from anonymised medical reports. This task was proposed by the 2010 edition of the I2B2/VA Natural Language Processing Challenges for Clinical Records [1]. Medical reports tend to be unstructured, arbitrarily expressed, and sometimes roughly thrown together, leading the NLP practitioner to deal with noisy documents.

Concept	Training	Test
Problem	7,073	12,592
Test	4,608	9,225
Treatment	4,844	9,344
Total	16,525	31,161

Table 1: Frequencies of concept types in the I2B2 2010 annotated corpus

BioCreative V CDR (BC5CDR) is a collection of 1,500 PubMed titles and abstracts selected from the CTD-Pfizer corpus and was used in the BioCreative V chemical-disease relation task. We use the standard training and test set in the BC5CDR shared task to extract the entities, and we do not perform entity linking.

3 Experimental Methodology

Transfer Learning Pre-trained word vectors have been an essential component in many NLP systems. Word representations are fed into a task-specific model, often improving the results. Re-

[1] https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/

Dataset	Training	Dev	Test
Disease	4,182	4,244	4,424
Chemical	5,203	5,347	5,385
Total	9,385	9,591	9,809

Table 2: Frequencies of entities in BC5CDR (chemical and disease) annotated corpus

cently, contextual word representations have significantly improved state of the art over non-contextual vectors. Transfer learning is leveraged here with the usage of BERT embeddings.

Domain Adaptation Adaptation can often take two forms: feature extraction, where the model's weights are used as-is as inputs of another system in a similar fashion to classic feature-based models and fine-tuning, where the model's weights continue to be trained on the new data for a specific task.

Graph Convolutional Networks Graph Convolutional Networks (GCNs) are often used for hierarchical representation problems. By performing convolution operations on neighbouring nodes (words) in the graph, a representation of a word will be enriched with information about its neighbours, which will allow the integration of information about the global context of the word. Since we are using the vocabulary to build the graph, we use VGCN (Lu et al., 2020). VGCN are able to take into account more global information about the vocabulary but often fail to capture some of the local information, which is why we use them combined with BERT. This paper considers lexical relations in a language, namely, the vocabulary graph, to be global information about the task-specific language. This vocabulary graph is constructed using both wordpieces (Wu et al., 2016) and word's co-occurrences alongside documents. We first select the relevant part of the global vocabulary graph according to the input token or sentence and transform it into an embedding representation. We then combine it with BERT token embedding and use multiple layers of attention mechanism to fuse the two.

The vocabulary graph is constructed using weighted positive point-wise mutual information (PPMI) (Levy and Goldberg, 2014). A higher PPMI indicate a higher semantic correlation between words. For each pair of words (w, c), we have:

$$P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$$

with the context probabilities raised to $\alpha = 0.75$, giving rare words are slightly higher probability.

$$PMI_\alpha(w,c) = max(log_2 \frac{P(w,c)}{P(w)P_\alpha(c)}, 0)$$

With this subword segmentation method, the word *epistaxis*, for example, will be represented as five tokens by BERT: *ep*, *##ista*, *##xi* and *##s*), since it is not in the model's training vocabulary. We are left with having to average or sum these embeddings to get a final embedding for our word. It puts us at a disadvantage when dealing with domain-specific corpora because most of the words in the area do not exist in the model's vocabulary. The model will rely heavier on the context to get embeddings for these wordpieces. We use both full words and wordpieces to build the vocabulary graphs in order to counteract this.

Given an undirected graph $G = (V, E)$ with a set N nodes $v_i \in V$, a set of edges (v_i, v_j), respectively, an adjacency matrix $A \in \mathbb{R}^{N \times N}$, a degree matrix $D_{ii} = \sum_j A_{ij}$ and a feature matrix $X \in \mathbb{R}^{N \times C}$, with C the number of dimensions of a feature vector, a forward 2-layer GCN model is computed as follow :

$$Z = f(X, A)$$

$$f(X, A) = softmax(\hat{A} \cdot ReLU(\hat{A}XW^0)W^{(1)})$$

with $\hat{A} = (\tilde{D}^{-\frac{1}{2}}\tilde{A})(\tilde{D}^{-\frac{1}{2}}X)$, the average of all neighbours feature vectors, scaled over both the rows and the columns of the matrix, putting more weights on low-degree nodes and reducing the impact of high-degree ones, and computed using $\tilde{A} = A + \lambda I_N$ (usually, $\lambda = 1$, but it can be treated as a trainable parameter) and $\tilde{D}$, its degree matrix. The adjacency matrix A corresponds to the weighted PPMI as the vocabulary graph and the feature matrix X to pre-trained embedding from BERT.

In the equation aforementioned, the GCN has two layers, as it is usually the standard (Kipf and Welling, 2017; Lu et al., 2020). With one layer GCN, each node can only get the information from its immediate neighbours. By adding another convolutional layer on top of it, we repeat the aggregation (or pooling process), but this time, the neighbours already have information about their

	I2B2			BC5CDR		
Model	P	R	F1	P	R	F1
BiLSTM-CRF	81.2 ± 0.4	84.4 ± 0.6	82.0 ± 0.3	78.2 ± 0.1	80.1 ± 0.8	79.2 ± 0.2
GCN	71.4 ± 0.5	52.1 ± 0.2	63.7 ± 0.1	79.9 ± 0.9	77.2 ± 1.0	78.6 ± 0.9
BERT	87.4 ± 0.3	87.0 ± 0.1	87.2 ± 0.8	86.0 ± 0.7	85.0 ± 0.8	85.5 ± 0.1
+ GCN$_{\text{vanilla}}$	87.7 ± 1.2	86.5 ± 0.2	87.4 ± 0.2	86.3 ± 0.6	**86.1 ± 0.7**	86.2 ± 0.3
+ GCN$_{\text{add}}$	**89.7 ± 0.4**	86.0 ± 0.7	87.7 ± 0.1	87.7 ± 0.4	85.7 ± 0.3	86.3 ± 0.2
+ GCN$_{\text{embedding}}$	89.0 ± 0.2	**88.8 ± 1.0**	**88.9 ± 0.2**	**87.9 ± 0.5**	85.7 ± 0.1	**86.7 ± 0.4**

Table 3: Analysis (in % F1-Score) of the outputs of our different models for the sequence labelling. This is an averaging of 3 runs for each experiment.

neighbours from the previous step. The number of layers is really the maximum number of hops that each node can reach to capture global information. However, we usually do not want to go too far in the graph. Otherwise, we may smooth out the graph, erasing important information entirely, making the representation less meaningful and resulting in a drop of performance (Kipf and Welling, 2017). Since the GCN's nodes are task entities, such as words, wordpieces or documents, this architecture requires all entities, including those from the training set, validation set, and test set, to be present in the graph during all the phases [2].

In the same fashion as (Lu et al., 2020), to combine these representations with BERT and to leverage both local and global information, we combine the vocabulary graph embedding obtained by our GCN and the BERT embedding and feed them to the first encoder. It will allow for the words' order in the sentence to be maintained and local information to be used, all the while the global information obtained by GCN will interact with BERT representation over the 12 layers of encoders. We also test two other combination methods: as for the first one, instead of integrating the GCN into the BERT embedding module, we simply add it to BERT embedding before passing it to the encoder. The second one consists of producing two outputs, one of the GCN and one of BERT, concatenating it just before applying a RELU and feeding it to the fully connected classification layer.

To summarise, multiple models are tested here, with several baselines in addition to the BERT and GCN combinations:

- **Bi-LSTM model**: BERT embeddings are used as input of a 256 hidden units and 2-layers bidirectional LSTM with an additional CRF layer.
- **GCN**: 2-layer GCN with BERT embeddings as input, with a simple fully-connected layer as output. This model only leverages global informations.
- **BERT**: pre-trained BERT for token classification, with a simple fully-connected layer as output. For all tasks, *bert-base-cased* is used.
- **BERT+GCN$_{\text{add}}$** or **BERT with added graph embedding**: Two representations are generated using BERT and GCN and are then summed. The combined representation is passed through a fully-connected layer for classification.
- **BERT+GCN$_{\text{embedding}}$** or **BERT with integrated graph embedding**: Instead of only using the regular BERT embeddings as input, we feed both the graph embedding obtained by the GCN and the BERT embedding to the BERT encoders.
- **BERT+GCN$_{\text{vanilla}}$** or **BERT with concatenated graph embedding**: Two representations are generated using BERT and GCN and are then concatenated. The combined representation is passed through a fully-connected layer for classification.

3.1 Pre-processings and Experimental settings

For all tasks, non-ASCII values, special characters and HTML tags are removed. Tokens from I2B2 and BC5CDR are then represented in the Inside–Outside–Beginning (IOB) tagging format for token classification. For the experimental settings,

[2] Masks are used during training to only use training nodes.

we implemented the models in PyTorch and PyTorch Geometric for the GCN part. All our experiments were run on a single GPU GEFORCE GTX 1080 for about ± 40 minutes per run (on average, over all the experiments).

4 Results

The results of the experiments are shown in Table 3. Performance is measured in macro-averaged scores (exact match). We report our results for the standard approaches first, and we contrast them with different combinations and architectures. Overall, all the BERT models with additional GCN global information perform better than the other baseline models, namely, the BiLSTM-CRF, the simple GCN and BERT. This confirms our intuitions and shows that it is beneficial to merge local and global information, and those resulting representations seem more worthwhile for the downstream tasks. A tendency seems to be showing (see Table 3): while getting a better F1-score, most of the boost in the overall score for the BERT with added global information goes to have better precision than the vanilla BERT and a lower recall simultaneously. This indicates an actual decrease in false positives, and these tendencies are similar across the board.

Future analysis can also be conducted on subwords since BERT breaks words down into wordpieces. The problem for specialised domains is that a more domain-oriented subword tokenisation method is probably more appropriate. For example, with BERT wordpieces tokeniser, *"adenocarcinoma"* will be broken into *"aden"*, *"oca"*, *"rc"*, *"ino"*, *"ma"* and, surprisingly, *"carcinoma"* will be broken into *"car"*, *"cino"*, *"ma"*, making *"ma"* the only subword that they share, even if the two words are semantically very close. Some methods have been developed, particularly for clinical text (Nguyen et al., 2019). However, using them means that we have to retrain the whole BERT model (similar to ClinicalBERT, for example), which defeats the purpose of adaptation. We rely on the GCN to fetch these missing pieces of information, connect them and integrate them back into the BERT model.

In this work, we wanted to examine if we can improve results for languages and domains for which there are no BioBERT and Clinical BERT, e.g. there are no WindBERT or PoliticalBERT for hypothetical wind energy or politics related tasks. This step of adaptation to the domain would make sense and even be necessary. Using a graph neural network constructed over the corpus' vocabulary exclusively follows the same logic. It comes from our decision to find a way to improve the model intrinsically without using external resources. Even if many resources exist for the English language, it is still crucial to explore ways to adapt existing models to our downstream specialised tasks where the volume of data is often insufficient. There is still a lot of room for improvement. One major limitation of this work — and work on graph neural networks in general, is the need to use all entities, including those from the training set, validation set, and test set, to build the vocabulary graph.

5 Conclusion

The quality of word representations obtained through language models is often correlated to the volume of data available. In the case of specialised domains, these methods might be less efficient due to the usually modest size of the corpora. This is particularly exacerbated in the case of specialised and low-resource domains, and the models might need to go through an adaptation phase. This work seeks to understand how these methods can be adapted on the fly by using additional features from GCN. The achieved results outperform the baselines across the board. It shows that it is beneficial to merge local and global information, and those resulting representations yield some additional advantages and are more worthwhile for the downstream task. Future work will cover analysis of attention layers before and after adding more global information. It will also involve considering more sophisticated relations than simple mutual information into the GCN, such as exploiting oriented graphs to encode dependencies, synonymy, hypo- and hyperonymy, and different architecture such as Graph Attention Networks.

Acknowledgements

This work has been funded by the French National Research Agency (ANR) and is under the ADDICTE project (ANR-17-CE23-0001).

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda B. Viégas, and Martin Wattenberg. 2019. Visualizing and Measuring the Geometry of BERT. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8594–8603. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'18)*, pages 4171–4186, Minneapolis, MN, USA.

Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger, and Jeremy Howard. 2019. MultiFiT: Efficient multi-lingual language model fine-tuning. 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 5702–5707, Hong Kong, China. Association for Computational Linguistics.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020. BERT-MK: Integrating graph contextualized knowledge into pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online. Association for Computational Linguistics.

Chanwoo Jeong, Sion Jang, Eunjeong Park, and Sungchul Choi. 2020. A context-aware citation recommendation model with bert and graph convolutional networks. *Scientometrics*, 124(3):1907–1922.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019. K-bert: Enabling language representation with knowledge graph.

Zhibin Lu, Pan Du, and Jian-Yun Nie. 2020. Vgcn-bert: Augmenting bert with graph embedding for text classification. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*, volume 12035 of *Lecture Notes in Computer Science*, pages 369–382. Springer.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Vincent Nguyen, Sarvnaz Karimi, and Zhenchang Xing. 2019. Investigating the effect of lexical segmentation in transformer-based models on medical datasets. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 165–171, Sydney, Australia. Australasian Language Technology Association.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, pages 1532–1543, Doha, Qatar.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP'19)*, volume abs/1903.05987, pages 7–14, Florence, Italy.

Timo Schick and Hinrich Schütze. 2020. BERTRAM: Improved word embeddings have big impact on contextualized model performance. pages 3996–4007.

Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. 2019. Pre-training of graph augmented transformers for medication recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5953–5959. International Joint Conferences on Artificial Intelligence Organization.

Balasubramaniam Srinivasan and Bruno Ribeiro. 2020. On the equivalence between positional node embeddings and structural graph representations. In *International Conference on Learning Representations*.

Ozlem Uzuner, Brett South, Shuying Shen, and Scott DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association : JAMIA*, 18:552–6.

Yonghui Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason R. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7370–7377.

Yukun Zhu, Ryan Kiros, Rich 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 *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 19–27, USA. IEEE Computer Society.