A Multi-task Approach for Named Entity Recognition in Social Media Data

Gustavo Aguilar, Suraj Maharjan, A. Pastor López-Monroy and **Thamar Solorio**
Department of Computer Science
University of Houston
Houston, TX 77204-3010
{gaguilaralas, smaharjan2, alopezmonroy, tsolorio}@uh.edu

Abstract

Named Entity Recognition for social media data is challenging because of its inherent noisiness. In addition to improper grammatical structures, it contains spelling inconsistencies and numerous informal abbreviations. We propose a novel multi-task approach by employing a more general secondary task of Named Entity (NE) segmentation together with the primary task of fine-grained NE categorization. The multi-task neural network architecture learns higher order feature representations from word and character sequences along with basic Part-of-Speech tags and gazetteer information. This neural network acts as a feature extractor to feed a Conditional Random Fields classifier. We were able to obtain the first position in the 3rd Workshop on Noisy User-generated Text (WNUT-2017) with a 41.86% entity F1-score and a 40.24% surface F1-score.

1 Introduction

Named Entity Recognition (NER) aims at identifying different types of entities, such as people names, companies, location, etc., within a given text. This information is useful for higher-level Natural Language Processing (NLP) applications such as information extraction, summarization, and data mining (Chen et al., 2004; Banko et al., 2007; Aramaki et al., 2009). Learning Named Entities (NEs) from social media is a challenging task mainly because (i) entities usually represent a small part of limited annotated data which makes the task hard to generalize, and (ii) they do not follow strict rules (Ritter et al., 2011; Li et al., 2012).

This paper describes a multi-task neural network that aims at generalizing the underneath rules of emerging NEs in user-generated text. In addition to the main category classification task, we employ an auxiliary but related secondary task called NE segmentation (i.e. a binary classification of whether a given token is a NE or not). We use both tasks to jointly train the network. More specifically, the model captures word shapes and some orthographic features at the character level by using a Convolutional Neural Network (CNN). For contextual and syntactical information at the word level, such as word and Part-of-Speech (POS) embeddings, the model implements a Bidirectional Long-Short Term Memory (BLSTM) architecture. Finally, to cover well-known entities, the model uses a dense representation of gazetteers. Once the network is trained, we use it as a feature extractor to feed a Conditional Random Fields (CRF) classifier. The CRF classifier jointly predicts the most likely sequence of labels giving better results than the network itself.

With respect to the participants of the shared task, our approach achieved the best results in both categories: 41.86% F1-score for entities, and 40.24% F1-score for surface forms. The data for this shared task is provided by Derczynski et al. (2017).

2 Related Work

Traditional NER systems use hand-crafted features, gazetteers and other external resources to perform well (Ratinov and Roth, 2009). Luo et al. (2015) obtain state-of-the-art results by relying on heavily hand-crafted features, which are expensive to develop and maintain. Recently, many studies have outperformed traditional NER systems by applying neural network architectures. For instance, Lample et al. (2016) use a bidirectional LSTM-

Proceedings of the 3rd Workshop on Noisy User-generated Text, pages 148–153

CRF architecture. They obtain a state-of-the-art performance without relying on hand-crafted features. Limsopatham and Collier (2016), who achieved the first place on WNUT-2016 shared task, use a BLSTM neural network to leverage orthographic features. We use a similar approach but we employ CNN and BLSTM in parallel instead of forwarding the CNN output to the BLSTM. Nevertheless, our main contribution resides on Multi-Task Learning (MTL) and a combination of POS tags and gazetteers representation to feed the network.

Recently, MTL has gained significant attention. Researchers have tried to correlate the success of MTL with label entropy, regularizers, training data size, and other aspects (Martínez Alonso and Plank, 2017; Bingel and Søgaard, 2017). For instance, Collobert and Weston (2008) use a multi-task network for different NLP tasks and show that the multi-task setting improves generality among shared tasks. In this paper, we take advantage of the multi-task setting by adding a more general secondary task, NE segmentation, along with the primary NE categorization task.

3 Methodology

This section describes our system[1] in three parts: feature representation, model description[2], and sequential inference.

3.1 Feature Representation

We select features to represent the most relevant aspects of the data for the task. The features are divided into three categories: character, word, and lexicons.

Character representation: we use an orthographic encoder similar to that of Limsopatham and Collier (2016) to encapsulate capitalization, punctuation, word shape, and other orthographic features. The only difference is that we handle non-ASCII characters. For instance, the sentence *"3rd Workshop !"* becomes *"ncc Ccccccc p"* as we map numbers to 'n', letters to 'c' (or 'C' if capitalized), and punctuation marks to 'p'. Non-ASCII characters are mapped to 'x'. This encoded representation reduces the sparsity of character features and allows us to focus on word shapes and punctuation patterns. Once we have an encoded word, we represent each character with a 30-dimensional vector (Ma and Hovy, 2016). We account for a maximum length of 20 characters[3] per word, applying post padding on shorter words and truncating longer words.

Word representation: we have two different representations at the word level. The first one uses pre-trained word embeddings trained on 400 million tweets representing each word with 400 dimensions (Godin et al., 2015)[4]. The second one uses Part-of-Speech tags generated by the CMU Twitter POS tagger (Owoputi et al., 2013). The POS tag embeddings are represented by 100-dimensional vectors. In order to capture contextual information, we account for a context window of 3 tokens on both words and POS tags, where the target token is in the middle of the window.

We randomly initialize both the character features and the POS tag vectors using a uniform distribution in the range $\left[-\sqrt{\frac{3}{dim}}, +\sqrt{\frac{3}{dim}}\right]$, where dim is the dimension of the vectors from each feature representation (He et al., 2015).

Lexical representation: we use gazetteers provided by Mishra and Diesner (2016) to help the model improve its precision for well-known entities. For each word we create a binary vector of 6 dimensions (one dimension per class). Each of the vector dimensions is set to one if the word appears in the gazetteers of the related class.

3.2 Model Description

Character level CNN: we use a CNN architecture to learn word shapes and some orthographic features at the character level representation (see Figure 1). The characters are embedded into a $\mathbb{R}^{d \times l}$ dimensional space, where d is the dimension of the features per character and l is the maximum length of characters per word. Then, we take the character embeddings and apply 2-stacked convolutional layers. Following Zhou et al. (2015), we perform a *global average pooling*[5] instead of the widely used *max pooling* operation. Finally, the result is passed to a fully-connected layer using a Rectifier Linear Unit (ReLU) activation function, which yields the character-based representation of

[1] https://github.com/tavo91/NER-WNUT17

[2] The neural network is implemented using Keras (https://github.com/fchollet/keras) and Theano as backend (http://deeplearning.net/software/theano/).

[3] Different lengths do not improve results

[4] http://www.fredericgodin.com/software

[5] Zhou et al. (2015) empirically showed that *global average pooling* captured more extensive information from the feature maps than *max pooling*.

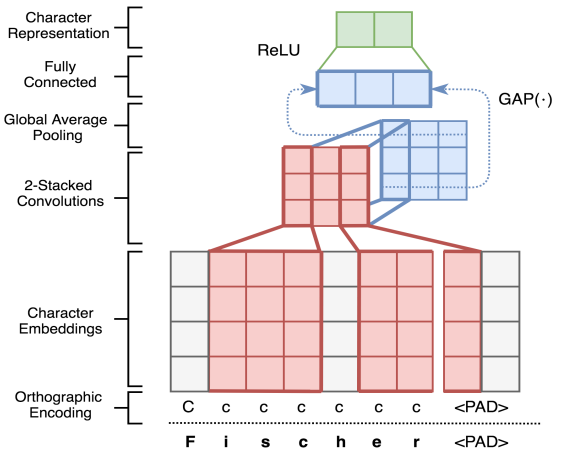

Figure 1: Orthographic character-based representation of a word (green) using a CNN with 2-stacked convolutional layers. The first layer takes the input from embeddings (red) while the second layer (blue) takes the input from the first convolutional layer. Global Average Pooling is applied after the second convolutional layer.

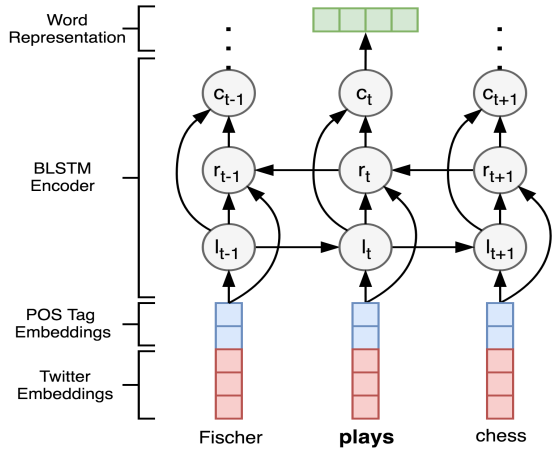

Figure 2: Word representation of POS-tag embeddings (blue) and Twitter word embeddings (red) using a BLSTM neural network.

a word. The resulting vector is used as input for the rest of the network.

Word level BLSTM: we use a Bidirectional LSTM (Dyer et al., 2015) to learn the contextual information of a sequence of words as described in Figure 2. Word embeddings are initialized with pre-trained Twitter word embeddings from a Skip-gram model (Godin et al., 2015) using word2vec (Mikolov et al., 2013). Additionally, we use POS tag embeddings, which are randomly initialized using a uniform distribution. The model receives the concatenation of both POS tags and Twitter word embeddings. The BLSTM layer extracts the features from both forward and backward directions and concatenates the resulting vectors from each direction ($[\vec{h}; \quad \overleftarrow{h}]$). Following Ma and Hovy (2016), we use 100 neurons per direction. The resulting vector is used as input for the rest of the network.

Lexicon network: we take the lexical representation vectors of the input words and feed them into a fully-connected layer. We use 32 neurons on this layer and a ReLU activation function. Then, the resulting vector is used as input for the rest of the network.

Multi-task network: we create a unified model to predict the NE segmentation and NE categorization tasks simultaneously. Typically, the additional task acts as a regularizer to generalize the model (Goodfellow et al., 2016; Collobert and Weston, 2008). The concatenation of character, word and lexical vectors is fed into the NE segmentation and categorization tasks. We use a single-neuron layer with a sigmoid activation function for the secondary NE segmentation task, whereas for the primary NE categorization task, we employ a 13-neuron[6] layer with a softmax activation function. Finally, we add the losses from both tasks and feed the total loss backward during training.

3.3 Sequential Inference

The multi-task network predicts probabilities for each token in the input sentence individually. Thus, those individual probabilities do not account for sequential information. We exploit the sequential information by using a Conditional Random Fields[7] classifier over those probabilities. This allows us to jointly predict the most likely sequence of labels for a given sentence instead of performing a word-by-word prediction. More specifically, we take the weights learned by the multi-task neural network and use them as features for the CRF classifier (see Figure 3). Taking weights from the common dense layer captures both of the segmentation and categorization features.

4 Experimental Settings

We preprocess all the datasets by replacing the URLs with the token <URL> before performing any experiment. Additionally, we use half of development set as validation and the other half as evaluation.

[6] Using BIO encoding, each of the 6 classes will have a *begin* and *inside* version (e.g. B-product, I-product).

[7] Python CRF-Suite library: https://github.com/scrapinghub/python-crfsuite

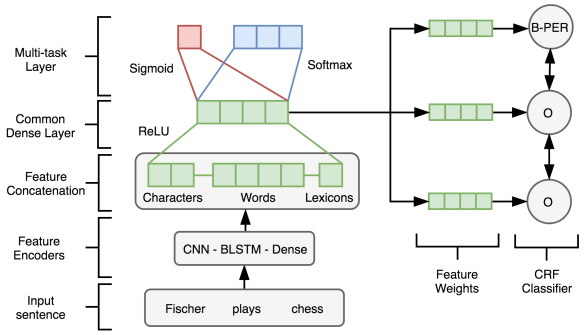

Figure 3: Overall system design. First, the system embeds a sentence into a high-dimensional space and uses CNN, BLSTM, and dense encoders to extract features. Then, it concatenates the resulting vectors of each encoder and performs multi-task. The top left single-node layer represents segmentation (red) while the top right three-node layer represents categorization (blue). Finally, a CRF classifier uses the weights of the common dense layer to perform a sequential classification.

Regarding the network hyper-parameters, in the case of the CNN, we set the kernel size to 3 on both convolutional layers. We also use the same number of filters on both layers: 64. Increasing the number of filters and the number of convolutional layers yields worse results, and it takes significantly more time. In the case of the BLSTM architecture, we add dropout layers before and after the Bidirectional LSTM layers with dropout rates of 0.5. The dropout layers allow the network to reduce overfitting (Srivastava et al., 2014). We also tried using a batch normalization layer instead of dropouts, but the experiment yielded worse results. The training of the whole neural network is conducted using a batch size of 500 samples, and 150 epochs. Additionally, we compile the model using the AdaMax optimizer (Kingma and Ba, 2014). Accuracy and F1-score are used as evaluation metrics.

For sequential inference, the CRF classifier uses L-BFGS as a training algorithm with L1 and L2 regularization. The penalties for L1 and L2 are 1.0 and $1.0e^{-3}$, respectively.

5 Results and Discussion

We compare the results of the multi-task neural network itself and the CRF classifier on each of our experiments. The latter one always shows the best results, which emphasizes the importance of sequential information. The results of the CRF, using the development set, are in Table 1.

Moreover, the addition of a secondary task allows the CRF to use more relevant features from the network improving its results from a F1-score of 52.42% to 54.12%. Our finding that a multi-task architecture is generally preferable over the single task architecture is consistent with prior research (Søgaard and Goldberg, 2016; Collobert and Weston, 2008; Attia et al., 2016; Maharjan et al., 2017).

Classes	Precision (%)	Recall (%)	F1 (%)
corporation	35.71	29.41	32.26
creative-work	60.00	5.26	9.68
group	30.00	12.00	17.14
location	65.71	56.10	60.53
person	83.98	62.04	71.36
product	39.29	15.71	22.45
Entity	72.16	43.30	54.12
Surface	68.38	95.05	79.54

Table 1: This table shows the results from the CRF classifier at the class level. The classification is conducted using the development set as both validation and evaluation.

Classes	Precision (%)	Recall (%)	F1 (%)
corporation	31.91	22.73	26.55
creative-work	36.67	7.75	12.79
group	41.79	16.97	24.14
location	56.92	49.33	52.86
person	70.72	50.12	58.66
product	30.77	9.45	14.46
Entity	57.54	32.90	41.86
Surface	56.31	31.31	40.24

Table 2: This table shows the final results of our submission. The hardest class to predict for is *creative-work*, while the easiest is *person*.

We also study the relevance of our features by performing multiple experiments with the same architecture and different combinations of features. For instance, removing gazetteers from the model drops the results from 54.12% to 52.69%. Similarly, removing POS tags gives worse results (51.12%). Among many combinations, the feature set presented in Section 3.1 yields the best results.

The final results of our submission to the WNUT-2017 shared task are shown in Table 2. Our approach obtains the best results for the *person* and *location* categories. It is less effective for *corporation*, and the most difficult categories for our system are *creative-work* and *product*. Our intuition is that the latter two classes are the most difficult to predict for because they grow faster and have less restrictive patterns than the rest. For instance, products can have any type of letters or numbers in their names, or in the case of creative works, as many words as their titles can hold (e.g.

Participants	F1 - E (%)	F1 - SF (%)
MIC-CIS	37.06	34.25
Arcada	39.98	37.77
Drexel-CCI	26.30	25.26
SJTU-Adapt	40.42	37.62
FLYTXT	38.35	36.31
SpinningBytes	40.78	39.33
UH-RiTUAL	**41.86**	**40.24**

Table 3: The scores of all the participants in the WNUT-2017 shared task. The metrics of the shared task are entity and surface form F1-scores. Our results are highlighted.

name of movies, books, songs, etc.).

Regarding the shared-task metrics, our approach achieves a 41.86% F1-score for entities and 40.24% for surface forms. Table 3 shows that our system yields similar results to the other participants on both metrics. In general, the final scores are low which states the difficulty of the task and that the problem is far from being solved.

6 Error Analysis

By evaluating the errors made by the CRF classifier, we find that the NE boundaries are a problem. For instance, when a NE is preceded by an article starting with a capitalized letter, the model includes the article as if it were part of the NE. This behavior may be caused by the capitalization features captured by the CNN network. Similarly, if a NE is followed by a conjunction and another NE, the classifier tends to join both NEs as if the conjunction were part of a single unified entity. Another common problem shown by the classifier is that fully-capitalized NEs are disregarded most of the time. This pattern may be related to the switch of domains in the training and testing phases. For instance, some Twitter informal abbreviations[8] may appear fully-capitalized but they do not represent NEs, whereas in Reddit and Stack Overflow fully-capitalized words are more likely to describe NEs.

7 Conclusion

We show that our multi-task neural network is capable of extracting relevant features from noisy user-generated text. We also show that a CRF classifier can boost the neural network results because it uses the whole sentence to predict the most likely set of labels. Additionally, our approach emphasizes the importance of POS tags in conjunction with gazetteers for NER tasks. Twitter word embeddings and orthographic character embeddings are also relevant for the task.

Finally, our ongoing work aims at improving these results by getting a better understanding of the strengths and weaknesses of our model. We also plan to evaluate the current system in related tasks where noise and emerging NEs are prevalent.

[8] E.g. *LOL* is an informal social media expression that stands for *Laughing Out Loud*, which is not an NE.

References

Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Mashuichi, and Kazuhiko Ohe. 2009. TEXT2TABLE: Medical Text Summarization System based on Named Entity Recognition and Modality Identification. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, BioNLP '09, pages 185–192, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mohammed Attia, Suraj Maharjan, Younes Samih, Laura Kallmeyer, and Thamar Solorio. 2016. CogALex-V Shared Task: GHHH - Detecting Semantic Relations via Word Embeddings. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*, pages 86–91, Osaka, Japan. The COLING 2016 Organizing Committee.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI'07, pages 2670–2676, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain. Association for Computational Linguistics.

Hsinchun Chen, Wingyan Chung, Jennifer Jie Xu, Gang Wang, Yi Qin, and Michael Chau. 2004. Crime Data Mining: A General Framework and Some Examples. *Computer*, 37(4):50–56.

Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. In *Proceedings of the 3rd Workshop on Noisy, User-generated Text (W-NUT) at EMNLP*. ACL.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-Based Dependency Parsing with Stack Long Short-Term Memory. *CoRR*, abs/1505.08075.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia Lab @ ACL WNUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, Beijing, China. Association for Computational Linguistics.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *CoRR*, abs/1502.01852.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. *CoRR*, abs/1603.01360.

Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012. TwiNER: Named Entity Recognition in Targeted Twitter Stream. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 721–730, New York, NY, USA. ACM.

Nut Limsopatham and Nigel Collier. 2016. Bidirectional LSTM for Named Entity Recognition in Twitter Messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 145–152, Osaka, Japan. The COLING 2016 Organizing Committee.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint Entity Recognition and Disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, Lisbon, Portugal. Association for Computational Linguistics.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *CoRR*, abs/1603.01354.

Suraj Maharjan, John Arevalo, Manuel Montes, Fabio A. González, and Thamar Solorio. 2017. A Multi-task Approach to Predict Likability of Books. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1217–1227, Valencia, Spain. Association for Computational Linguistics.

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? Semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53, Valencia, Spain. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Shubhanshu Mishra and Jana Diesner. 2016. Semi-supervised Named Entity Recognition in noisy-text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 203–212, Osaka, Japan. The COLING 2016 Organizing Committee.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, Georgia. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Learning Deep Features for Discriminative Localization. *CoRR*, abs/1512.04150.