XD at SemEval-2020 Task 12: Ensemble Approach to Offensive Language Identification in Social Media Using Transformer Encoders

Xiangjue Dong
Computer Science
Emory University
Atlanta, GA, USA
xiangjue.dong@emory.edu

Jinho D. Choi
Computer Science
Emory University
Atlanta, GA, USA
jinho.choi@emory.edu

Abstract

This paper presents six document classification models using the latest transformer encoders and a high-performing ensemble model for a task of offensive language identification in social media. For the individual models, deep transformer layers are applied to perform multi-head attentions. For the ensemble model, the utterance representations taken from those individual models are concatenated and fed into a linear decoder to make the final decisions. Our ensemble model outperforms the individual models and shows up to 8.6% improvement over the individual models on the development set. On the test set, it achieves macro-F1 of 90.9% and becomes one of the high performing systems among 85 participants in the sub-task A of this shared task. Our analysis shows that although the ensemble model significantly improves the accuracy on the development set, the improvement is not as evident on the test set.

1 Introduction

With the development of IT, social media has become more and more popular for people to express their views and exchange ideas publicly. However, some people may take advantage of the anonymity in social media platform to express their comments rudely, and attack other people verbally with offensive language. To keep a healthy online environment for the adolescences (Chen et al., 2012) and to filter offensive messages for the users (Razavi et al., 2010), it is necessary and significant for technology companies to develop an efficient and effective computational methods to identify offensive language automatically.

Transformer-based contextualized embedding approaches such as BERT (Devlin et al., 2019a), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020) or ELECTRA (Clark et al., 2020) have re-established the state-of-the-art for many natural language classification tasks especially the GLUE Dataset (Wang et al., 2018). Their pre-trained models were pre-trained on different large datasets, for example, BERT was pre-trained on the BOOKCORPUS (Zhu et al., 2015) and English Wikipedia, and RoBERTa was pre-trained on CC-NEWS (Nagel, 2016), OPENWEBTEXT (Gokaslan and Cohen, 2019), and STORIES (Trinh and Le, 2018) which enable their models to learn different language features.

This paper presents six transformer-based offensive language identification models that learn different features from the target utterance. To combine the distinctive learned language features, we introduce an ensemble strategy which concatenates the representations of the individual models and feed them into the linear decoder to make binary classification (Section 4.2). It largely improves the performance over the baseline on our dev set (Section 4.4).

2 Related Work

Offensive language in Twitter (Wiegand et al., 2018), Facebook (Kumar et al., 2018), and Wikipedia (Georgakopoulos et al., 2018) has been widely studied. In addition, different aspects of offensive language have been studied, like the type and target of offensive posts (Zampieri et al., 2019), cyberbullying (Dinakar et al., 2011; Huang et al., 2014), aggression (Kumar et al., 2018), toxic comments (Georgakopoulos et

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2244–2250
Barcelona, Spain (Online), December 12, 2020.

al., 2018) and hate speech (Badjatiya et al., 2017; Davidson et al., 2017; Malmasi and Zampieri, 2017; Malmasi and Zampieri, 2018).

Many deep learning approaches have been used to address the task. The Convolutional Neural Networks (CNNs), Long Short-Term Memory Networks (LSTMs) and FastText were applied on the hate speech detection task (Badjatiya et al., 2017). Gamback and Sikdar (2017) used four Convolutional Neural Network (CNN) models with random word vectors, word2vec word vectors, character n-gram, and concatenation of word2vec word embeddings and character n-grams as feature embeddings separately to categorize each tweet into four classes: racism, sexism, both (racism and sexism) and non-hate-speech.

3 Data Description

The datasets we use are Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019) and Semi-Supervised Offensive Language Identification Dataset (SOLID) (Rosenthal et al., 2020). Given a tweet, the task is to predict whether the content involves offensive language. Table 1 shows the examples of offensive and non-offensive tweets in these two datasets.

Id	Tweet	Label
09	@USER Buy more icecream!!!	NOT
71	@USER That's because you are an old man.	OFF

(a) Examples from OLID.

Id	Tweet	AVG_CONF	CONF_STD
167	@USER Pre-ordered your book, received in July, started last night and cannot put it down!	0.215	0.188
524	a combination of innocence and corruption	0.691	0.142

(b) Examples from SOLID.

Table 1: Examples in OLID and SOLID. NOT: not offensive, OFF: offensive, AVG_CONF: average of the confidences to be offensive, CONF_STD: confidences' standard deviation

OLID is a collection of 14,100 English tweets annotated as OFF or NOT. It is divided into a training set of 13,240 tweets and test set of 860 tweets (Zampieri et al., 2019). SOLID is a collection of about 9 million English tweets labeled in a semi-supervised manner (Rosenthal et al., 2020). The data are annotated with AVG_CONF and CONF_STD predicted by several supervised models (Zampieri et al., 2020). The test set provided by organizers this year has 3887 tweets. Table 2 shows the statistics of OLID and SOLID.

	OLID	SOLID
TRN	13240	9089140
TST	860	3887

Table 2: Statistics of OLID and SOLID. TRN: training set, TST: test set.

4 Experiments

4.1 Data Split

For our experiments, a combination of OLID and SOLID (Section 3) is used. We find that about 1.0% of SOLID are duplicates, which have been removed before data splitting. For the dataset used for fine-tuning classification model, we set threshold of AVG_CONF (Section 3) to be 0.5 in SOLID, which means the data with AVG_CONF above 0.5 is labelled as OFF. 90% of the TRN of OLID is combined with the whole SOLID as the new training set TRN for default transformer-based models fine-tuning (FT). The remaining 10% of the TRN and the TST of OLID is used as the development set DEV of FT. All the existed datasets are combined together as the training set TRN for model pre-training (PT). After pre-training, 99.5% of the SOLID is randomly selected as the training set TRN and 0.5% of the SOLID is randomly selected to create the development set DEV for fine-tuning our pre-trained models into classification models and regression models (PT-C and PT-R). In PT-C, the data with AVG_CONF above 0.5 is labelled as OFF and in PT-R, original value of AVG_CONF is used. Furthermore, 90% of TRN in OLID is randomly

selected as the new training set TRN, and 10% of TRN in OLID is combined with the TST of the OLID and become the development set DEV for classification models and regression models' further fine-tuning (PT-C-C and PT-R-C). The ensemble model is fine-tuned on the same dataset as PT-C-C. Table 3 shows the detailed statistics of the data split in our experiments.

	FT	PT	PT-R	PT-C	PT-R-C	PT-C-C	E
TRN	8,963,663	9,107,127	8,951,747	8,951,747	11,916	11,916	11,916
DEV	2,184	-	44,983	44,983	2,184	2,184	2,184

Table 3: Statistics of the data split used for our experiments. TRN: training set, DEV: development set, FT: dataset used for default model fine-tuning, PT: dataset used for default model pre-training, PT-R: dataset used for fine-tuning our pre-trained models into regression models, PT-C: dataset used for fine-tuning our pre-trained models into classification models, PT-R-C: dataset used for fine-tuning regression model into classification models, PT-C-C: dataset used for further fine-tuning classification models, E: dataset used for fine-tuning ensemble models.

4.2 Models

In general, default transformer-based models are fine-tuned as baseline models. The sequence of embeddings of input generated from the transformer encoder is fed into linear decoder to gain the output vector that makes the binary classification. Then we pre-train these default models and choose the models with lowest perplexity. Next, we fine-tune the pre-trained models into regression models and classification models based on corresponding dataset, respectively. Furthermore, the regression models and classification models are fine-tuned again into classification models. In the end, sentence presentation of individual models are concatenated and fed into linear decoder to generate the output vector that makes the binary decision of whether or not this tweet is offensive.

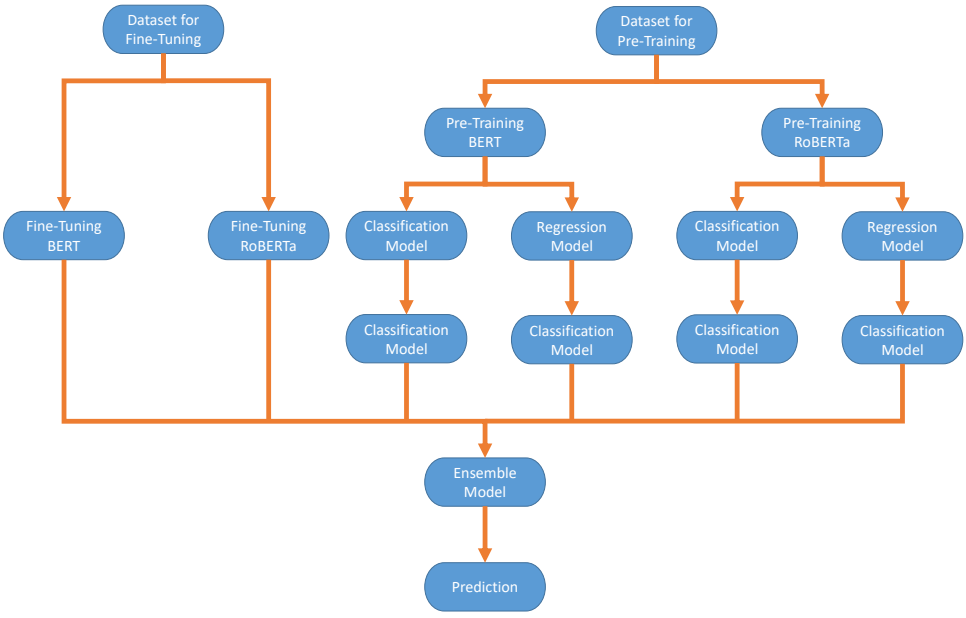

Figure 1: Overview of the individual models and the ensemble model

In our experiments, two types of transformer-based models are used as the default models, BERT-Base model (Devlin et al., 2019b) and RoBERTa-Base model (Liu et al., 2020). For the default model fine-tuning part, the default BERT-Base and RoBERTa-Base are fine-tuned on FT (Section 4.1) as baseline models. For the pre-training part, the BERT-Base and RoBERTa-Base are pre-trained on PT (Section 4.1). Then, the two pre-trained models which have the lowest perplexity are fine-tuned into regression models and classification models separately on PT-R and PT-C. Next, the fine-tuned pre-trained models are further fine-tuned into classification models on PT-R-C and PT-C-C. Finally, sentence presentation of

six individual models are concatenated to form the ensemble model which is fine-tuned on E. Figure 1 shows the overview of the six individual models and the ensemble model.

4.3 Experimental Setup

According to our experiments, the data preprocessing doesn't contribute significantly to the final prediction results on such huge dataset. Thus, we skip the data preprocessing. According to the analysis of sentence length in the dataset, we set *max_length* of the models to be 128. After an extensive hyper-parameter search, we set *learning_rate* to be $2e-5$, *seed_value* to be 42, and *epochs* to be 10 for our six individual models and ensemble model. After that, we also experiment more on the ensemble model and find that the best result is gained by changing *learning_rate* to $1e-5$ and *dropout* to 0.5.

4.4 Results

Table 4 shows the results achieved by our individual models and ensemble model. The selected pre-trained BERT-base model and pre-trained RoBERTa-base model have the lowest perplexities, which are 21.3 and 47.5. Our fine-tuned pre-trained classification-classificaion BERT and RoBERTa models outperform their counterpart baseline by about 1.7% and 1.1%, respectively. In addition, our fine-tuned pre-trained regression-classification BERT and RoBERTa models show 2.1% and 1.8% improvements over their baselines. The ensemble model with *learning_rate* of $1e-5$ and *dropout* of 0.5 (E_2) achieves significantly improvement on development set. It outperforms the BERT baseline and RoBERTa baseline by 8.5% and 8.6%, respectively. As a result, we use this ensemble model as our final model and submit the prediction results to the shared task's CodaLab page.[1] We achieve a macro-F1 score of 90.901% on the test set and rank 36th among 85 participants in sub-task A. After the release of the gold labels, we also calculate our other models' performance on test set (Table 4) and make detailed comparison and analysis among them (Section 4.5.1).

Model	ACC_DEV	ACC_TST	P_TST	R_TST	F1_TST	Epochs
B-FT	83.784	**92.153**	88.990	94.510	**90.933**	6
R-FT	83.692	92.102	88.933	94.503	90.882	10
B-PT-C-C	85.204	90.610	88.402	88.115	88.256	1
B-PT-R-C	85.845	92.102	88.933	94.532	90.885	2
R-PT-C-C	84.654	92.102	88.933	94.532	90.885	1
R-PT-R-C	85.158	88.552	85.129	87.886	86.299	2
E	88.548	**92.153**	88.990	94.510	**90.933**	2
E_1	90.701	**92.153**	88.992	94.396	90.917	1
E_2	**90.884**	92.128	88.962	94.464	90.901	2

Table 4: Results of individual models and ensemble model on dev set and test set. B-FT: fine-tuned default BERT-base, R-FT: fine-tuned default RoBERTa-base, B-PT-C-C: fine-tuned our pre-trained BERT-base classification-classification model, R-PT-C-C: fine-tuned our pre-trained RoBERTa-base classification-classification model, B-PT-R-C: fine-tuned our pre-trained BERT-base regression-classification model, R-PT-R-C: fine-tuned our pre-trained RoBERTa-base regression-classification model, E: ensemble model with default *learning_rate* of $2e-5$, E_1: ensemble model with lower *learning_rate* of $1e-5$, E_2: submitted ensemble model with higher *dropout* of 0.5.

4.5 Analysis

4.5.1 Ablation Analysis

When we fine-tuned our pre-trained models, B-PT-C, B-PT-R, R-PT-C, and R-PT-R on only 10% of the PT-R and PT-C (Section 4.4) separately, the accuracy of models, B-PT-C-C, B-PT-R-C, R-PT-C-C, and R-PT-R-C we get is 82.822%, 83.326%, 83.280%, and 83.646%, which is lower than the results using total data (Table 4). It indicates that deep learning models which are trained on larger dataset perform better. For the ensemble model, when we decrease the *learning_rate* from $2e-5$ (E) to $1e-5$ (E_LL), the performance improves from 88.548% to 90.701%, which shows that the ensemble

[1] https://competitions.codalab.org/competitions/23285

model is sensitive to the change in learning rates. By changing the default *dropout* from 0.1 (E_LL) to 0.5 (E_HD), the model performance increase to 90.884%, which indicates the influence of the dropout rate. After comparing the predicted labels from our unsubmitted models with the released gold labels (Table 4), we can see the model which achieves the highest accuracy on the development set doesn't perform best on the test set. which may be caused by overfitting. Pure fine-tuned BERT-base model (B_FT) achieves the same accuracy as other two ensemble models. In addition, higher accuracy can't guarantee the higher f1-score due to the data imbalance.

4.5.2 Error Analysis

The confusion matrix in Figure 2 further displays the error pattern of our classifier on test set. As we can see, there are only three instances labeled with OFF are misclassified to NOT while more data labeled with NOT are classified to OFF. Table 5 shows these three misclassified offensive examples and other misclassified not offensive tweets.

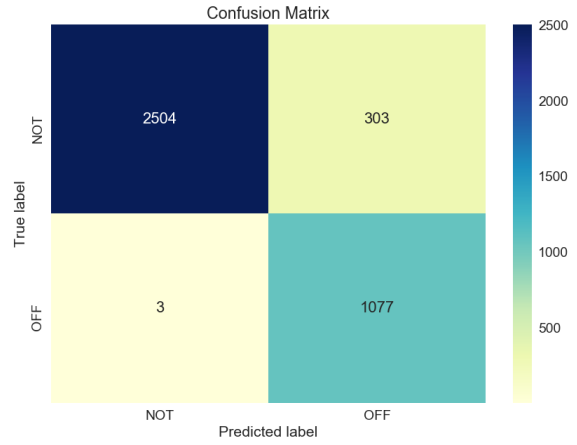

Figure 2: Confusion matrix of the ensemble model

One explanation of the results may be that the imbalance of the dataset leads to the classifier's preference for the majority class. It is possible that our classifier may not capture some of the subtle nuances in meaning and contexts, and our system still needs some improvement for these subtle details.

Id	Tweet	PL	TL
304	Can someone please jump her ass.	NOT	OFF
2333	@USER I don't So far as you can recognize your Dad aa trash there's no need to be talking to you.	NOT	OFF
2825	@USER Wings over and it's not even a question (sweet chili & Jamaican jerk hanger).	NOT	OFF

(a) Misclassified offensive examples.

Id	Tweet	PL	TL
3564	@USER @USER @USER Do not engage with idiots, they'll bring you down to their level and beat you with experience.	OFF	NOT
3725	This heartburn is disgusting.	OFF	NOT

(b) Misclassified not offensive examples.

Table 5: Misclassified examples. PL: predicted label, TL: true label

4.6 Conclusion

This paper explores the performance of six individual transformer-based models and their ensemble model for the task of offensive language identification in social media. Default BERT-Base and RoBERTa-Base individual fine-tuning models are adapted to establish the strong baselines for the ensemble model. Sentence representations from six individual models are concatenated and fed into the linear decoder to make binary decision for the ensemble model. Our ensemble model with higher dropout shows significant improvements on accuracy, up to 8.6%, on the dev set than baseline models. However, it performs worse than the baseline model B-FT and original ensemble model E on the test set, which has a 92.153%

accuracy. It may be caused by model overfitting and data imbalance, which are the problems we need to take into consideration in future experiments.

Acknowledgments

We gratefully acknowledge the support of the AWS Machine Learning Research Awards (MLRA). Any contents in this material are those of the authors and do not necessarily reflect the views of AWS.

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Learning for Hate Speech Detection in Tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, page 759–760, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Y. Chen, Y. Zhou, S. Zhu, and H. Xu. 2012. Detecting Offensive Language in Social Media to Protect Adolescent Online Safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Conference on Social Computing*, pages 71–80, Sep.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *International Conference on Learning Representations*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *International AAAI Conference on Web and Social Media*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

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

Karthik Dinakar, Roi Reichart, and Henry Lieberman. 2011. Modeling the Detection of Textual Cyberbullying. In *International AAAI Conference on Web and Social Media*.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-Speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, Vancouver, BC, Canada, August. Association for Computational Linguistics.

Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, and Vassilis P. Plagianakos. 2018. Convolutional Neural Networks for Toxic Comment Classification. *CoRR*, abs/1802.09957.

Aaron Gokaslan and Vanya Cohen, 2019. *OpenWebText Corpus*.

Qianjia Huang, Vivek Kumar Singh, and Pradeep Kumar Atrey. 2014. Cyber Bullying Detection Using Social and Textual Analysis. In *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia*, SAM '14, page 3–6, New York, NY, USA. Association for Computing Machinery.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In *Proceedings of the International Conference on Learning Representations*.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. In *RANLP*.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30(2):187–202.

Sebastian Nagel, 2016. *News Dataset Available*.

Amir H. Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive Language Detection Using Multi-level Classification. In Atefeh Farzindar and Vlado Kešelj, editors, *Advances in Artificial Intelligence*, pages 16–27, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Weakly Supervised Dataset for Offensive Language Identification. In *arxiv*.

Trieu H. Trinh and Quoc V. Le. 2018. A Simple Method for Commonsense Reasoning. *arXiv*, 1806.02847.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November. Association for Computational Linguistics.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria – September 21, 2018*, pages 1 – 10. Austrian Academy of Sciences, Vienna, Austria.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32*, pages 5754–5764. Curran Associates, Inc.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the Type and Target of Offensive Posts in Social Media. 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 1415–1420, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

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 *The IEEE International Conference on Computer Vision (ICCV)*, December.