DLJUST at SemEval-2021 Task 7: Hahackathon: Linking Humor and Offense

Hani Al-Omari
Jordan University of Science and Technology
Computer Information Systems Department
Irbid, Jordan
alomarihani1997@gmail.com

Isra'a AbedulNabi
Jordan University of Science and Technology
Computer Information Systems Department
Irbid, Jordan
israha95@gmail.com

Rehab Duwairi
Jordan University of Science and Technology
Computer Information Systems Department
Irbid, Jordan
rehab@just.edu.jo

Abstract

Humor detection and rating poses interesting linguistic challenges to NLP; it is highly subjective depending on the perceptions of a joke and the context in which it is used. This paper utilizes and compares transformers models; BERT base and Large, BERTweet, RoBERTa base and Large, and RoBERTa base irony, for detecting and rating humor and offense. The proposed models, where given a text in cased and uncased type obtained from SemEval-2021 Task7: HaHackathon: Linking Humor and Offense Across Different Age Groups. The highest scored model for the first subtask: Humor Detection, is BERTweet base cased model with 0.9540 F1-score, for the second subtask: Average Humor Rating Score, it is BERT Large cased with the minimum RMSE of 0.5555, for the fourth subtask: Average Offensiveness Rating Score, it is BERTweet base cased model with minimum RMSE of 0.4822.

1 Introduction

SemEval 2021 Task7 is constructed to detect and rate the humor and offense inside jokes in the English language (Meaney et al., 2021). Humor is an essential aspect of strengthening human communication and relations. However, the interpretation of humor differs based on the perceptions of a joke and the context in which it is used. In 2012, the Human Rights Commission found the most commonly reported form of harassment in Australia was sexist or offensive jokes(the, 2012), humor appreciation; is a highly subjective phenomenon as a sense of humor varies from person to person depending on factors such as age, gender, and socio-economic status. In this task, data labels and ratings were collected from a balanced set of age groups from 18-70. Moreover, annotators represent a variety of genders, political stances, and income levels.

The automatic detection of linguistic elements in natural language texts such as aggression, humor, irony, and sarcasm has drawn attention to research communities (Davidov et al., 2010). Several studies and experiments have been performed to develop and improve humor detection systems(Annamoradnejad, 2021)(Winters and Delobelle, 2020)(Sane et al., 2019)(Mao and Liu, 2019)(Chen and Soo, 2018). BERT language model(Annamoradnejad, 2021)(Devlin et al., 2019) showcase the highest results compared to all other works. This paper aims to utilize the Transformers models; BERT base and Large (Devlin et al., 2019), BERTweet (Nguyen et al., 2020), RoBERTa base and Large (Liu et al., 2019), and RoBERTa base irony (Barbieri et al., 2020) for humor detection, humor rating, and offense rating using the dataset obtained from SemEval-2021 Task7 that contains training, development, and test data. Our contributions are: Prepossessing text techniques for text tokenization, word segmentation, spell correction, removing the punctuation, encoding, and extracting embeddings. Furthermore, training six state-of-the-art Transformers Models and compare its results against the Base-line Model. We have achieved a 0.9540 F1-score for Subtask1-A Humor Detection using BERTweet cased model compared to the first place score, which is 0.9820 F1-score. For Subtask1-B Average Humor Score, our RMSE result is 0.5555 using BERT Large cased model, the first place RMSE is 0.4959. Finally, for Subtask2 Average Offensiveness Score, our RMSE results is 0.4822 using BERTweet cased model, while the first place RMSE is 0.4120.

This paper's structure is as follows: Section 2 reviews related works focused on Humor detection in SemEval-2020. Section 3 presents data exploration, prepossessing, training models, and evaluation metrics. Section 4 introduces the experiments and results. Section 5 remarks the conclusion and proposes future works.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1114–1119
Bangkok, Thailand (online), August 5–6, 2021.

2 Related Work

The previous Humor Detection task on SemEval2020 Task7 was focused on humor rating only without considering if it is offensive or not. However, in SemEval2021 Task7, the task requires detecting the hidden offense inside jokes. Rozental et al., 2020 (Rozental et al., 2020) presented a novel L2-Regularization approach with freezing the weights for the first epoch to train and fine-tune the word embedding model, ensemble different language models - BERT, XL-NET, and Roberta, and duplication from each language models, with a weighted average between them. Their approach ranked second place in SemEval-2020 Task 7: "Assessing Humor in Edited News Headlines", subtasks 1 and 2.
Shatnawi et al., 2020 (Shatnawi et al., 2020) also proposed the BERT-Flair-based Humor Detection Model (BFHumor) that combined the BERT regressor and Flair library to predict the funniest values of edited headlines for the same Task 7 of SemEval 2020; the mode ranked 4th in subtask1 and 12th in the subtask2. Meanwhile, Pramodith Ballapuram 2020 (Ballapuram, 2020) participated in the same task using a non-ensemble model; he proposed a Siamese Transformer based approach, coupled with an Attention mechanism to make use of contextual embeddings and focus words and their impact against other tokens on generating important features and rating the funniness of the edited headline, he scored fifth place in subtask1 and fourth place in subtask2.

3 Methodology

Our methodology of tackling the humor detection problem consists of four phases: Data exploration and Visualization, Data Pre-Processing, Learning Models, and Evaluation Criteria.

3.1 Data Exploration

The dataset from SemEval-2021 Task7: HaHackathon: Detecting and Rating Humor and Offense, consists of five columns as table 1 shows; col-1 is the id of the text, col-2 "text" is the raw text for a joke to process, col-3 "is-humor" is a binary classification for the text, 1 means it is humor and 0 means it is not humor, col-4 "humor-rating" is a numerical representation for how much humorous is the text if it is labeled as humor from col-2, col-5 "humor-controversy" is binary classification to represent the subjectivity of humor appreciation with a controversy score, 1 means the humor of the text is controversial and 0 is not, col-6 "offense-rating" is a numerical representation for how much offensive is the text. The competition consists of two subtasks: subtask1 is divided into three parts A,B and C and predicts is-humor, humor-rating, and humor-controversy respectively. Subtask2 is to predict the offense-rating.

Data sections are described separately, starting with subtask1-A; it is a binary classification problem to detect whether the text is humor or not. The distribution between its classes is balanced, so no need for data upsampling or downsampling. Subtask1-B is dependent on subtask1-A; if the text is labeled as humorous, a value will be provided to humor rating, and if it is not, the rating will be none. For this task, we dropped records that are labeled as not humorous. Humor rating is a regression problem since the rating is a continuous value between zero and five, the values are distributed normally which helps the model to generalize better. Subtask1-C also depends on subtask1-A; if the text is classified as humorous, then predict if the humor rating would be considered controversial. It is a binary classification task, and the distribution between its classes is balanced. Table 2 shows the number of instances per each class for both "is_humor" and "humor_controversy". Subtask2 is a regression problem to predict how offensive a text would be, the target value is between zero and five. After checking the target distribution, it is skewed to the left, which indicates the model will have difficulty in training, reducing the chances of predicting the values above 3.

We explored the number of the words distribution per instance and the number of unique words; where we do not count the same words. Both distributions were similar in density, that indicated most of the text content are unique and non-repetitive words. the maximum number of words within instances is 70 words for the whole data set and the average number of words is around 20 words. We choose to define the input sequence length equal to 128 as through tokenization, some words will be divided to multiple tokens. To check odd or very long words that need to be handled, we visualize the mean word length for each instance in the data set, and the word length is in the average mean. Finally, we checked the number of punctuations in the text since it affects the model as it affects the text in the encoding phase, especially when it is

id	text	is_humor	humor_rating	humor_controversy	offense_rating
1	TENNESSEE: We're the best state. Nobody even comes close. *Elevennessee walks into the room* TENNESSEE: Oh shit...	1	2.42	1	0.2
297	I met a vaping vampire from Romania. He called himself Vlad the Inhaler.	1	2.05	0	0
4698	What's the difference between black people and cancer? Cancer got Jobs.	1	1.75	0	4.2
5231	Fellas: Don't be mad when someone else starts to appreciate the woman you took for granted. What you won't do, someone else will.	0			0.3
5300	Black people love boom boxes .. I hate to generalize, but it's their stereotype ;-)	1	1.54	0	2.9

Table 1: Train dataset

feature name	negative	positive
is_humor	3068	4932
humor_controversy	2467	2465

Table 2: Number of instances that belong to each class for "is_humor" and "humor_controversy"

attached with a word (e.g., animals do not encode the same as animal's). We handle the punctuation in the preprocessing part, which will be described in the next section.

3.2 Data Pre-Processing

In this phase, we apply enhancement techniques to the text. Removing duplicate sentences, repetitive characters, spilling mistakes, and stop words. Also, it includes encoding methodology to transform text from its original form to a vector that makes the computer understands it. We used Ekphrasis (Baziotis et al., 2017) for spell correction, remove contraction words, and annotate caps text as it is crucial to know the speaker's tone, capital letters indicate a level of aggression. We used GloVe (Pennington et al., 2014) vocabulary to find the out of vocabulary words that Ekphrasis did not fix, applied some spell correction manually, and removed the punctuation that has been attached to some words using regular expressions. We applied the tokenization technique. For GloVe embeddings, using Keras tokenization tool that splits the tokens based on the space. For example ["We went to Aqaba."] will be tokenized as the following: ['We', 'went',' to',' Aqaba.'], and each token gets encoded. On the other hand, the BERT model uses a WordPiece tokenizer that depends on its own vocabulary, and if it faces an out-of-vocabulary word, it will be split into sub tokens that starts with ##token. For example, ['tokenization'] become ['token','##ization'], and for the RoBERTa model, it uses byte pair encoding (BPE) word pieces. RoBERTa handles the out-of-vocabulary words the same way as the BERT model but with some modification on the algorithm. For example: ['tokenization'] become ['token',' ization']. We encoded the text using GloVe and transformers; Glove considers frequency when building the embeddings, unlike word2vec.

3.3 Learning Models

This section will describe the models we have used; it will be divided into two sub-sections: Baseline Model, and Transformers Models.

3.3.1 Base Line Model

The baseline model is BiLSTM model. It takes an encoded sentience with 100 token sequence size as input, each encoded using GloVe embeddings that have been fed into the embedding layer; which considered to be as a lookup table which consists of 300-dimensional pretrained GloVe embeddings, and each row in the table is considered a representation of the word. Next is two BiLSTM models that consist of 128 nodes, 0.2 dropouts to avoid overfitting, and He uniforms weight initializer (He et al., 2015). Output passes through a feed-forward network which consist of four hidden layers of 512, 256, 128 and 64 neurons respectively, for each layer we use ReLU activation function, 0.4 dropouts, and He uniforms weight initializer. Finally, the output layer consists of the Sigmoid activation function in binary classification and linear layer for the Regression task. We used Stochastic Gradient Descent (SGD) with a 0.01 learning rate, 0.99 momentum, and Nesterov implementation (Nesterov, 2003). It is worth mentioning that we used the early stopping technique to avoid overfitting with five patience.

3.3.2 Transformers Models

We have applied different type of pretrained models using Simple Transformers library (Rajapakse) which is an API built above Hugging Face library (Wolf et al., 2019). In this section, we generally describe the models that we used. Bidirectional Encoder Representation from Transformers (BERT) is a pretrained model which uses attention models to learn the contextual relation between the words in the sentence, consisting of two main parts: an encoder and a decoder: an encoder that encodes the text, and a decoder for the output result based on the task. We used Bert cased and uncased models, which both have been trained on BookCoupus (Zhu et al., 2015) with 800 million words and En-

glish Wikipedia with 2,500 million words. We trained the model using this hyperparameter: 128 sequence length, three epochs, 32 batch size, 4e-5 learning rate, and AdamW as an optimizer. Then we used the BERTweet model trained on BERT architecture using 850 million English tweets. We trained this model using almost the same hyperparameters, except that we used eight as batch size. We have used, too, Robustly Optimized BERT pretrained approach (RoBERTa). It is a fine-tuned version of the BERT model with some changes on the data size and input representation. Training the model on larger data set significantly improves the model performance using BookCurpus, English Wikipedia, CC-news with 63 million English news articles, OpenWebText, and Stories; which is a subset of CommonCrawl data. Model developers use dynamic masking instead of static masking that has been used in the BERT model, this technique allows to improve the model performance. Furthermore, they have used the full sentence without using the next sentence prediction loss. We trained both RoBERTa base and large models using the following hyperparameters: for the base three 128 sequence length, three epochs, 32 batch size, 4e-5 learning rate, and AdamW optimizer, and the large model, we kept everything the same as the base model but change the batch size to 16. Moreover, we have used the RoBERTa irony model that has been fine-tuned using 58 million tweets for irony detection on TweetEval benchmark; also, we used everything as the base and large model. We have also used XLM-RoBERTa (Conneau et al., 2020), and XLNet (Yang et al., 2019) as a black box, we did not go into the model's detail, but in general, it is an improved version model from BERT. We applied them using the following hyperparameters: 128 sequence length, three epochs, 16 batch size, 4e-5 learning rate, and AdamW optimizer.

3.4 Evaluation Criteria

F1 score criteria was used for the binary classification task and the second criteria is RMSE for the regression task. We used the 8-Fold Cross-validation method to determine the best model since we only have an 8k data instances for training. We trained our models using seven folds, kept the last fold unseen for validation in each iteration, and used every model from each iteration to predict the development and testing data and combine all the results to obtain the final result.

4 Experimentation and Results

4.0.1 Task 1-A Is-Humor:

We constructed a baseline model and fine-tuned it using different approaches and prepossessing techniques. We tested our model using four types of preprocessing techniques (None, Ekphrasis, Ekphrasis with removing stop words, and Ekphrasis with applying Custom Spell Correction). After that, we tested Adam and SGD as opimizers, SGD performs better on this model. After comparison, the best parameters for the models are described in the fifth row of table 3; we applied the early stopping technique to reduce the overfitting with 0.4 dropout. We test Transformers models, refer to the table 4. In general, we fine-tuned all the models in two ways, the Cased model, which means that the text is kept in its original form without lowering the characters' case. Uncased means that lower-case all the characters in the text. By experiment, cased models performed better than uncased since it captures more aggressive behaviors from the writer. Moreover, BERTweet performs well using the Cased model on this task. Since the model has already been trained on Twitter data, it captures all the slang, acronyms, and abbreviations.

4.0.2 Task 1-B: Humor Rating

We used the experiments from the previous task since they are dependent if humor equals zero; we do not need to predict the humor rating. If it is one, we have to predict the humor rating value between zero and five. We first used the best model from the baseline by changing the last layer to a linear layer to predict continuous values since it is a regression task. Same hyperparameters from model 5 from table 3 we got 0.8651 RMSE, and we consider it as our baseline. After that, we tested the best transformer models from the previous task, refer to table 5. We found out that the best model is the BERT large case model, which seems unexpected since most of the other models perform almost the same, around 0.54.

4.0.3 Task 2 Offensiveness Score:

We have used the same methodology from the Subtask1-B; we first constructed a baseline for this task, which is the best model from the Subtask 1-A hyperparameters model 5 from table 3, and we got 0.86783 RMSE. After that, we tested the best models from the previous task's transformers, refer to table 6. We found out that the best model is BERT

	Pre-Processing	learning rate	# epochs	dropout	batch size	optimizer	features	Accuracy	Precision	Recall	F1-score
1	None	0.0001	50	.4	128	Adam	GloVe	0.8594	0.8778	0.8968	0.8872
2	Ekph + remove stopwords	0.0001	50	.4	128	Adam	GloVe	0.8535	0.8804	0.8821	0.8813
3	Ekph	0.0001	50	.4	128	Adam	GloVe	0.875	0.9083	0.8867	0.8974
4	Ekph + Custom Spell-Correction	0.0001	50	.4	128	Adam	Glove	.88062	0.9066	.8990	0.9028
5	Ekph + Custom Spell-Correction	0.0001	50	.4	128	SGD	Glove	0.8806	0.9003	0.9067	0.9035

Table 3: Baseline model experiments on is-humor task

	Model	Text Type	# epochs	batch size	Accuracy	Precision	Recall	F1-score
1	BERT base uncased	Uncased	3	32	0.9424	0.9515	0.9552	0.9533
2	BERT Large uncased	Uncased	3	32	0.9455	0.9582	0.9532	0.9556
3	**BERTweet uncased**	**Uncased**	**3**	**8**	**0.9561**	**0.9679**	**0.9607**	**0.9643**
4	RoBERTa base	Uncased	3	32	0.9448	0.9593	0.9548	0.9570
5	RoBERTa Large	Uncased	3	16	0.8366	0.8366	0.9686	0.8978
6	RoBERTa base irony	Uncased	3	32	0.9494	0.9574	0.9607	0.9590
7	XLM-RoBERTa large	Uncased	3	16	0.7408	0.7087	0.9837	0.8239
8	XLNet base	Uncased	3	16	0.9449	0.9572	0.9531	0.9551
9	XLNet large	Uncased	3	16	0.8030	0.7750	0.9588	0.8571
10	BERT base cased	Cased	3	32	0.9440	0.9558	0.9532	0.9545
11	BERT Large cased	Cased	3	32	0.9505	0.9597	0.9600	0.9599
12	**BERTweet cased**	**Cased**	**3**	**32**	**0.9589**	**0.9704**	**0.9627**	**0.9665**
13	RoBERTa base	Cased	3	32	0.9494	0.9615	0.95612	0.9588
14	**RoBERTa Large**	**Cased**	**3**	**32**	**0.9563**	**0.9702**	**0.9584**	**0.9643**
15	RoBERTa base irony	Cased	3	16	0.9543	0.9633	0.9625	0.9629

Table 4: Transformers models on is-humor task

Model	Text Type	# epochs	batchsize	RMSE
BERTweet	Uncased	3	8	0.5479
RoBERTa base	Uncased	3	32	0.5417
BERT base	Cased	3	32	0.5360
BERT Large	**Cased**	**3**	**32**	**0.5296**
BERTweet	Cased	3	32	0.54585
RoBERTa base	Cased	3	32	0.54272
RoBERTa Large	Cased	3	32	0.54548
RoBERTa irony	Cased	3	16	0.54585

Table 5: Transformers models on humor rating

Large model and it performs well on regression tasks since it performs well on the two tasks related to regression. All the previous experiments applied the cross-validation method on the training data. Since we decided that we do not want to overfit the development and test dataset, we will use the best models that perform well on the cross-validation phase to predict the development and test dataset's output. We changed the threshold point for the is-humor task; using the ROC curve, and the best split is 0.233357 since it improves the model by 0.04 percent on the development phase, so we apply it to the testing phase. The best model for the is humor task is BERTweet cased base model that scored a 0.9540 F1 score on the testing phase; for the humor rating task, we used BERT large cased model that scored 0.5555 RMSE on the testing phase, and for the offensive score, we used BERTweet large cased model that scored 0.4822 RMSE on testing phase.

Model	Text Type	# epochs	batchsize	RMSE
BERTweet	Uncased	3	8	0.5304
RoBERTa base	Uncased	3	32	0.5734
RoBERTa Large	Uncased	3	32	0.5494
BERT base	Cased	3	32	0.5516
BERT Large	**Cased**	**3**	**32**	**0.5247**
BERTweet	Cased	3	32	0.5302
RoBERTa base	Cased	3	32	0.5522
RoBERTa Large	Cased	3	32	0.7190
RoBERTa irony	Cased	3	16	0.8501

Table 6: Transformers models on offensiveness score

5 Conclusion and Future Work

In this paper, we experimented with a set of state-of-the-art Transformers and contextual models for detecting and rating humor and offense in text. Our experimental results show that the BERTweet Large model is the best model for humor binary classification task with a 0.9540 F1 score and offensive rating with 0.4822 RMSE, and BERT Large cased model is the best for humor rating task scored 0.5555 RMSE. We plan to enhance the top models using ensemble learning methodology and test out more novel methods.

References

2012. Working without fear: Results of the sexual harassment national telephone survey.

Issa Annamoradnejad. 2021. Colbert: Using bert sentence embedding for humor detection.

Pramodith Ballapuram. 2020. Lmml at semeval-2020 task 7: Siamese transformers for rating humor in edited news headlines. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1026–1032.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Peng-Yu Chen and Von-Wun Soo. 2018. Humor recognition using deep learning. 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 113–117.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcasm in twitter and amazon. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 107–116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

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.

Jihang Mao and Wanli Liu. 2019. A bert-based approach for automatic humor detection and scoring. In *IberLEF@ SEPLN*, pages 197–202.

J.A. Meaney, Steven R. Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. Semeval 2021 task 7, hahackathon, detecting and rating humor and offense. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.

Yurii Nesterov. 2003. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Thilina Rajapakse. [link].

Alon Rozental, Dadi Biton, and Ido Blank. 2020. Amobee at semeval-2020 task 7: Regularization of language model based classifiers. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 981–985.

Sushmitha Reddy Sane, Suraj Tripathi, Koushik Reddy Sane, and Radhika Mamidi. 2019. Deep learning techniques for humor detection in hindi-english code-mixed tweets. In *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 57–61.

Fara Shatnawi, Malak Abdullah, and Mahmoud Hammad. 2020. Mlengineer at semeval-2020 task 7: Bert-flair based humor detection model (bfhumor). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1041–1048.

Thomas Winters and Pieter Delobelle. 2020. Dutch humor detection by generating negative examples.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

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*, pages 5753–5763.

Yukun Zhu, Ryan Kiros, Richard 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.