Unifying Different NLP Tasks with A Question-answering Model

Stanford CS224N Default Project

Polycarpos Yiorkadjis
Department of Computer Science
Stanford University
polys@stanford.edu

Yvonne Wang
Department of Computer Science
Stanford University
yvonnew@stanford.edu

Abstract

In this project, we present a new question-answering (QA) based model to unify three NLP tasks, i.e., sentiment analysis (SA), paraphrase detection (PD) and semantic textual similarity (STS). Unlike the traditional multitask framework which uses independent heads to handle different tasks and therefore has limitations on solving unseen tasks, our model simultaneously performs three tasks with a simple unified QA head by adding one or several questions to describe the goal and requiring the model to handle the tasks through answering the corresponding question. Experimental results show that this QA based unified model can achieve superior performance on both dev sets (SA: 52.4%, PD: 88.6%, STS: 85.3%) and test sets (SA: 52.4%, PD: 88.7%, STS: 85.3%), which are comparable against the competitive multitask baseline with multiple heads.

1 Key Information to include

- Mentor: Shai Limonchik
- External Collaborators (if you have any): None
- Sharing project: None

2 Introduction

As introduced in the default project [1], we aim to design an approach to finetuning BERT (Devlin et al., 2019) model so that it can simultaneously perform three given tasks: sentiment analysis, paraphrase detection, and semantic textual similarity. There is a classical idea of the multitask framework which uses multiple heads to handle different tasks. However, this solution is not a unified model for different NLP tasks, because if we want to handle additional tasks, it may need to add more independent heads. Since different tasks may require diverse head designs, this solution has limitations on building a general-purpose AI model. For instance, text classification problems, relation detection problems and regression problems usually have different heads, which limits the generation ability of this model on unseen tasks.

According to previous works (Kumar et al., 2016; McCann et al., 2018; Khashabi et al., 2020), most tasks in NLP can be unified under one framework by casting them as triplets of the question, context, and answer. In our view, such question-answering (QA) based unified model has three advantages: 1) The task specific design is no longer needed; 2) Diverse datasets with different annotations can be directly used to train one model; 3) The model trained by this unified QA framework has the potential to handle unseen tasks.

In this project, we propose a QA based unified model on top of BERT to simultaneously handle these three tasks. As shown in the right part of Figure 1, for each task, in addition to its original input,

[1] https://web.stanford.edu/class/cs224n/project/default-final-project-bert-handout.pdf

we add one or several questions to describe the goal and require the model to handle this task by answering the corresponding question. We find that, with reasonable question settings, the unified head of this model can be very simple, where only one linear transformation combined with a sigmoid function is used to output a number in [0,1] so that the model can answer both "Yes/No" and the similarity score. Experimental results show that, even with such a single head, the proposed QA based unified model can achieve comparable accuracy on three tasks against a competitive baseline with multiple heads.

The main contributions of this work are as follows:

- We build a strong baseline with multiple heads to simultaneously handle three tasks, i.e., sentiment analysis (SA), paraphrase detection (PD), and semantic textual similarity (STS). This model has achieved superior performance on both dev sets (SA: 51.8%, PD: 89.2%, STS: 85.8%, ranked 8th) and test sets (SA: 52.7%, PD: 89.0%, STS: 86.4%, ranked 5th).
- We further propose a QA based model to unify the three NLP tasks mentioned above with a single unified head. This QA based unified model can also achieve excellent performance on both dev sets (SA: 52.4%, PD: 88.6%, STS: 85.3%) and test sets (SA: 52.4%, PD: 88.7%, STS: 85.3%), which are comparable against the competitive baseline.
- Additional ablation studies are executed to help analyze several designs in our models.

3 Related Work

Recently, there is a trend to unify different NLP tasks under a same framework and train a unified model to achieve state-of-the-art results on different tasks. These works can be roughly categorized into three types. The first type of works, e.g., (Kumar et al., 2016; McCann et al., 2018; Khashabi et al., 2020), tried to unify different tasks by casting them as triplets of the question, context, and answer and proposed corresponding QA based models. For the second type, some previous works, e.g., GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2020) and GPT-3 (Brown et al., 2020), introduced a unified framework that converts all text-based language problems into a text-to-text format so that the model could handle different tasks with semantically meaningful prompts. Other than the above formulations, some recent works represented by ChatGPT (Ouyang et al., 2022) unified NLP tasks through the way of dialogue. With suitable prompts, we think GPT or T5 models may also perform quite well on these three tasks. However, these models are very large and expensive. If we just would like to develop a model to solve problems in a small range of practical scenarios, designing a small but effective unified model can be a good choice. Therefore, the work in this project is also meaningful in real-world applications.

4 Approach

In this work, we propose two approaches to handle three tasks with one single model, i.e., 1) a multitask baseline model with multiple heads, and 2) a QA based unified model. The details of these two models will be introduced in the following subsections.

4.1 Multitask Baseline with Multiple Heads

As shown in the left part of Figure 1, our multitask baseline is to add three independent heads for different tasks which take the feature $F_{cls} \in R^d$ of [CLS] token extracted by BERT model as input and output the corresponding predictions. For different tasks, we design different input formats and heads.

Sentiment Analysis (SA). This task requires our model to classify each input sentence into several categories. We set the input format to [CLS] + input_sentence + [SEP]. The SA head will first perform a dropout operation on the feature of [CLS] token and then predict the logits for different categories with a fully-connected (FC) layer. These logits are activated by a softmax function. Let K denotes the number of category and $W_{SA} \in R^{d \times K}$ denotes the weights of FC layer, the output O_{SA} of SA head can be calculated as follows:

$$O_{SA} = Softmax(W_{SA}(Dropout(F_{cls}))) \tag{1}$$

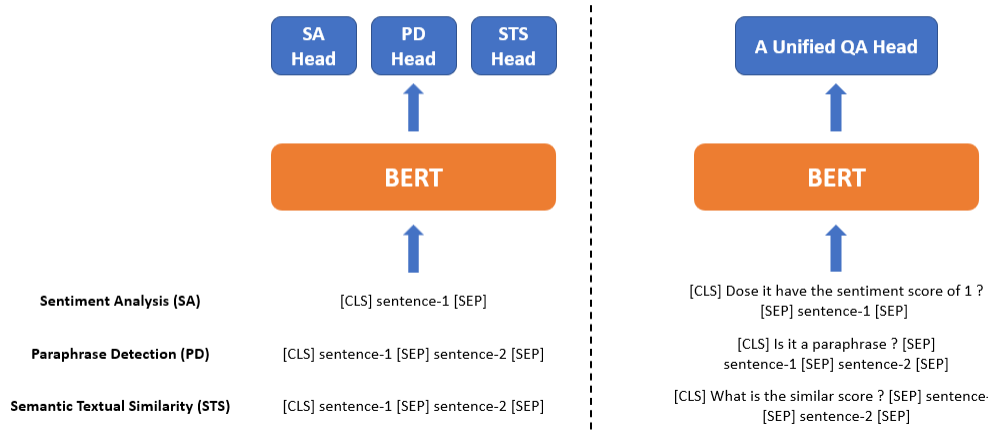

Figure 1: An overview of our baseline with multiple heads (left) and QA based unified model (right).

Paraphrase Detection (PD). We formulate this task as a binary classification task. Since the input contains two sentences, we use the input format: [CLS] + input_sentence_1 + [SEP] + input_sentence_2 + [SEP]. The head is almost the same as the SA head and the only difference is that the output is a single logit and activated by a sigmoid function. Let $W_{PD} \in R^{d \times 1}$ denotes the weights of FC layer, the output O_{PD} of SA head can be calculated as follows:

$$O_{PD} = Sigmoid(W_{PD}(Dropout(F_{cls}))) \tag{2}$$

Semantic Textual Similarity (STS). We formulate this task as a regression problem. The input format and the head design are the same as those of the PD task. Since the regression targets in SemEval STS (Agirre et al., 2013) is between 0 and 5.0, we multiply the output value of sigmoid function by 5. Let $W_{STS} \in R^{d \times 1}$ denotes the weights of FC layer, the output O_{STS} of SA head can be calculated as follows:

$$O_{STS} = Sigmoid(W_{STS}(Dropout(F_{cls}))) * 5 \tag{3}$$

4.2 Question-answering based Unified Model

As shown in the right part of Figure 1, we add addition questions to the input of different tasks and use a unified head to handle all the three tasks.

Input Formats. For the SA tasks, the formats will be [CLS] + question + [SEP] + input_sentence + [SEP]. Since the label of this task contains multiple categories, we generate several questions for each sentence and the format of questions is "Does it have the sentiment score of #i?", where $i \in \{0, 1, 2, ..., K\}$ and K is the number of categories. The model will run K times and output the category with the highest score. For the PD and STS task, the format will be [CLS] + question + [SEP] + input_sentence_1 + [SEP] + input_sentence_2 + [SEP]. The question is set to "Is it a paraphrase?" for the PD task and "What is the similarity score?" for the STS task. The model will only run once to handle each sentence pair in these two tasks.

Question-answering Head. The design of the unified QA head is quite simple but effective. This head takes the feature of [CLS] token extracted by BERT model as input. We first perform a dropout operation on the input and then use a fully-connected (linear transformation) layer to output a single logit. This logit will be activated by a sigmoid function to output a number in [0,1] so that it can answer both "Yes/No" with a confidence score and the similarity score. While training, the ground-truth labels used for the STS task are normalized into [0,1]. Let $W_{QA} \in R^{d \times 1}$ denotes the weights of FC layer, the output O_{QA} of SA head can be calculated as follows:

$$O_{QA} = Sigmoid(W_{QA}(Dropout(F_{cls}))) \tag{4}$$

4.3 Loss Functions

During training, each training sample in three tasks will appear once in each epoch and we implement a special data loader to make the training samples in each batch come from the same task so that only the corresponding head is trained in this batch. Therefore, we only introduce the loss function used in each task.

4.3.1 Multitask Baseline

Sentiment Analysis Head. While training, we use the cross-entropy loss in this head. The loss L_{SA} is as follows:

$$L_{SA} = CrossEntropy(O_{SA}, O^*_{SA}) \tag{5}$$

where O_{SA} is the output of SA head in each mini-batch and O^*_{SA} is the corresponding labels.

Paraphrase Detection Head. The loss function used in this head is the binary-cross-entropy loss. Therefore, the loss L_{PD} in this head is calculated by:

$$L_{PD} = BinaryCrossEntropy(O_{PD}, O^*_{PD}) \tag{6}$$

where O_{PD} is the output of PD head in each mini-batch and O^*_{PD} denotes the corresponding labels.

Semantic Textual Similarity Head. We formulate the STS task as a regression problem and apply L1 loss in this head. The loss L_{STS} is as follows:

$$L_{STS} = |O_{STS} - O^*_{STS}| \tag{7}$$

where O_{STS} is the output of STS head in each mini-batch and O^*_{STS} denotes the corresponding ground-truth similarity scores.

4.3.2 QA based Unified Model

For the SA and PD tasks, we use binary-cross-entropy loss to train the QA based unified head. Let O_{QA} denotes the output feature of QA head, the loss L_{QA-SA} and L_{QA-PD} can be calculated as follows:

$$L_{QA-SA} = BinaryCrossEntropy(O_{QA}, GT_{SA}) \tag{8}$$

$$L_{QA-PD} = BinaryCrossEntropy(O_{QA}, GT_{PD}) \tag{9}$$

where GT_{SA} and GT_{PD} denote the ground-truth labels for the SA and PD tasks, respectively.

For the STS task, we also use the L1 Loss in our multitask baseline to train the QA head. The loss L_{QA-STS} is as follows:

$$L_{QA-STS} = |O_{QA} - GT_{STS}| \tag{10}$$

where GT_{STS} denotes the corresponding ground-truth similarity scores.

5 Experiments

5.1 Data

We conduct experiments on Stanford Sentiment Treebank dataset (Socher et al., 2013) for setiment analysis, Quora dataset [2] for paraphrase detection, and SemEval STS dataset (Agirre et al., 2013) for semantic textual similary.

Stanford Sentiment Treebank (SST) contains 8,544 training, 1,101 validating, and 2,210 testing sentences extracted from movie reviews. The dataset was parsed with the Stanford parser [3] and includes a total of 215,154 unique phrases from those parse trees, each annotated by 3 human judges Each phrase has a label of negative, somewhat negative, neutral, somewhat positive, or positive.

[2] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[3] https://nlp.stanford.edu/software/lex-parser.shtml

Table 1: Experimental results of different models on the dev sets and test sets.

Is QA	Train			Dev				Test			
	SST Dataset	Quora Dataset	SemEval Dataset	SST Accurary	Quora Accurary	SemEval Pearson score	Overall Score	SST Accurary	Quora Accurary	SemEval Pearson score	Overall Score
	✓			0.510	-	-	-	-	-	-	-
		✓		-	0.887	-	-	-	-	-	-
			✓	-	-	**0.865**	-	-	-	-	-
	✓	✓	✓	0.518	**0.892**	0.858	**0.756**	**0.527**	**0.890**	**0.864**	**0.760**
✓	✓	✓	✓	**0.524**	0.886	0.853	0.754	0.524	0.887	0.853	0.755

Quora consists of 400,000 question pairs with labels indicating whether particular instances are paraphrases of one another. We use a subset of this dataset with 141,506 training, 20,215 validating, and 40,431 testing question pairs.

SemEval STS consists of 6,041 training, 864 validating, and 1,726 testing sentence pairs. The annotation is the similarity score of each pair of sentences on a scale from 0 (unrelated) to 5 (equivalent meaning).

5.2 Evaluation Methods

Following the default project, we use the accuracy as evaluation metric for the sentiment analysis and the paraphrase detection tasks and calculate the Pearson correlation of the true similarity values against the predicted similarity values across the test dataset as in Agirre et al. (2013) as the evaluation metric for the semantic texutal similarity task.

5.3 Experimental Details

All experiments are implemented in Pytorch v1.8.0 and conducted on a workstation with 1 Nvidia Tesla V100 GPU. We use BERT-Base in all experiments and its weights are initialized with a pre-trained model. The models are optimized by AdamW Loshchilov and Hutter (2017) algorithm with batch size 16. We use a fixed learning rate 1e-5 and the betas, epsilon of AdamW are set as (0.9, 0.999) and 1e-6, respectively. All the dropout rates in different heads are 0.3. We train all the models by 10 epochs. For multitask training, each training sample in three datasets will appear once in each epoch and we implement a special data loader to make the training samples in each batch come from the same dataset so that only the corresponding head is trained in this batch. For the unified QA model, we use $K = 5$ in SST dataset and generate five training samples for each sentence. To balance the overall loss of postive samples and negative samples in SST dataset, the loss weight of positive samples and negative samples in the unified QA model are set as 4 and 1, respectively. For inference, the threshold used in the paraphrase detection task is 0.5 in all the models.

5.4 Results

5.4.1 Comparisons with Baselines

We first compare the results of our multitask baseline with multiple heads on three dev sets with three models which are only trained and evaluated on a single task with the corresponding head. As shown in Table 1, this baseline has achieved the results of 51.8%, 89.2% and 85.8% on the SST, Quora and SemEval datasets respectively, which are comparable against the other three models on each task (SST: 51.0%, Quora: 88.7% and SemEval: 86.5%). This small gap within experimental variance demonstrates the effectiveness of our designs and training strategies. By the way, we do not observe obvious performance gain after training model on more data samples (one dataset v.s. three datasets). We think the reason may be that the data source and labels of these three datasets are quite diverse so that it is not easy for the BERT model to learn some common information.

We then compare the results of our QA based unified model with the multitask baseline on both dev sets and test sets. The QA model has achieved an overall score of 75.4% on dev sets and 75.5% on test sets, which are comparable against the overall score of 75.6% and 76.0% of the multitask baseline model on dev and test sets, respectively. According to our observation, the gap between these two models are within the experimental variance, so we have successfully unified these three tasks with a simple unified QA head. It should be noted that, for fair comparisons, all the hyper-parameters used in these experiments are the same.

Table 2: Ablation study of different loss weight settings while training the unified QA model on STS dataset.

Loss Weight (Pos. : Neg.)	Dev SST Accurary	Dev Quora Accurary	Dev SemEval Pearson score	Dev Overall Score
1:1	0.510	0.884	**0.862**	0.752
2:1	0.516	**0.886**	0.852	0.751
4:1	**0.524**	**0.886**	0.853	**0.754**

Table 3: Ablation study of adding more fully-connected (FC) layers combined with RELU activation to the unified QA head.

# of Additional FC Layers (Linear(768, 768) + RELU)	Dev SST Accurary	Dev Quora Accurary	Dev SemEval Pearson score	Dev Overall Score
0	**0.524**	0.886	0.853	0.754
1	0.519	**0.888**	0.857	**0.755**
2	0.510	0.884	**0.862**	0.752

5.4.2 Ablation Studies

To further analyze the effectiveness of some model designs and hyper-parameters in our QA based unified model, we have executed several groups of ablation studies on the dev sets. The details will be introduced in the following subsections.

Effectiveness of the Loss Weight Settings while Training our QA Model on SST Dataset. While training our QA based unified model on SST dataset, since the category number of this dataset is five, we generate five training samples with slightly different questions for each input sentence. In each group of five samples, there will be one positive sample and four negative samples. Therefore, directly set the loss weight of all training samples as 1 may encounter an imbalanced issue. To address this problem, we set the loss weight of positive samples as 4 and the loss weight of negative samples as 1. The experimental results in Table 2 show that this strategy can improve the accuracy on SST by 1.4% absolutely compared with the loss weight setting of 1:1.

Ablation Study of Adding more Fully-connected (FC) Layers. As mentioned in the approach section, we only add a single FC layer in the QA based unified head to directly output the prediction logit. Here, we have also tried to add more additional FC layers combined with RELU activation after the dropout operation and feed the output of the last additional FC layer to the original FC layer for prediction. As shown in Table 3, we find that adding more FC layers can not significantly improve the performance. Therefore, we set the number of additional FC layer as zero by default. These results also demonstrate the effectiveness of our question (prompt) designs so that only a very simple QA head can work very well.

Ablation Study of Different Weight Decay Settings. The default setting of weight decay in this project is zero. However, we observe that the weight decay is usually set as 0.01 while finetuning BERT-style pretrained language models in some previous works like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Therefore, we have executed additional experiments with different weight decay settings and the results are in Table 4. We surprisingly find that setting weight decay as zero can achieve the best performance, which does not meet our expectation. Maybe the reason is that fitting these three datasets at the same time is quite challenging so that we do not need to add weight decay to avoid overfitting.

Ablation Study of Different Learning Rate Settings. We also execute additional experiments with different learning rates. As shown in Table 5, setting learning rate as 1e-5 has achieved the best performance among all these three settings. Therefore, we set the learning rate as 1e-5 by default.

6 Discussion

In this section, we would like to discuss about two points we have noticed while doing this project. The details are introduced in the following subsections.

Table 4: Ablation study of different weight decay settings.

Weight decay	Dev			
	SST Accurary	Quora Accurary	SemEval Pearson score	Overall Score
0.0	**0.524**	0.886	**0.853**	**0.754**
0.001	0.511	**0.887**	0.852	0.750
0.01	0.496	0.884	0.849	0.743

Table 5: Ablation study of different learning rate settings.

learning rate	Dev			
	SST Accurary	Quora Accurary	SemEval Pearson score	Overall Score
5e-6	0.516	0.877	0.838	0.744
1e-5	**0.524**	**0.886**	**0.853**	**0.754**
2e-5	0.509	0.885	0.834	0.743

6.1 Relatively Low Accuracy on the Sentiment Analysis Task

We notice that the accuracy of our models on the sentiment analysis task is relatively lower than those on the other two tasks. After analyzing the failure cases on the dev set of SST dataset, we find that this task is really more challenging than the other two because there are many ambiguous cases. For instance, for the movie review "It's a lovely film with lovely performances by Buy and Accorsi.", the ground-truth is "neutral" while the model predicts "somewhat positive". In our view, the model also predicts a reasonable answer. Actually, different people may have different answers, so two adjacent scores may be both correct in fact (e.g., neutral (3) v.s. somewhat positive (4) in the above example). According to this observation, we further analyze the predictions and find that for about 86.3% failure cases, the gap between the ground-truth label and predicted label is only 1. Therefore, the capabilities of our model are underestimated on the sentiment analysis task.

6.2 Generation Ability on Unseen Tasks

The proposed QA based model has the potential to handle unseen tasks when given suitable questions. However, the current version is still far away from this goal. The main reasons come from two aspects: 1) The number of tasks and training samples are not large enough, which will limit the generation ability of this model. 2) The questions designed for training are not diverse. If we want the model to handle unseen tasks by answering corresponding questions, the question formats and contents should be more diverse so that the model can easily understand the semantic meanings of new questions.

7 Conclusion

In this project, we presented a new question-answering based model to unify three NLP tasks, i.e., sentiment analysis, paraphrase detection and semantic textual similarity. Compared to the traditional multitask framework which uses independent heads to handle different tasks, our QA based unified model can perform different tasks with a simple unified head by answering the corresponding questions, which has the potential to show better generation ability on unseen tasks. Experimental results show that this QA based unified model can achieve comparable performance against the competitive multitask baseline with multiple heads on Standford Sentiment Treebank, Quora and SemEval STS datasets. In the future work, we would like to train this QA based unified model on more NLP tasks with more training samples. Furthermore, we plan to explore how to generate more diverse questions and have a thoroughgoing evaluation on unseen tasks.

References

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational*

*Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR.

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. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.