Revisiting the Importance of Encoding Logic Rules in Sentiment Classification

Kalpesh Krishna♠♣ **Preethi Jyothi**♠ **Mohit Iyyer**♣

Indian Institute of Technology, Bombay♠
University of Massachusetts, Amherst♣
{kalpesh,miyyer}@cs.umass.edu
pjyothi@cse.iitb.ac.in

Abstract

We analyze the performance of different sentiment classification models on syntactically-complex inputs like *A-but-B* sentences. The first contribution of this analysis addresses reproducible research: to meaningfully compare different models, their accuracies must be averaged over far more random seeds than what has traditionally been reported. With proper averaging in place, we notice that the distillation model described in Hu et al. (2016), which incorporates explicit logic rules for sentiment classification, is ineffective. In contrast, using contextualized ELMo embeddings (Peters et al., 2018a) instead of logic rules yields significantly better performance. Additionally, we provide analysis and visualizations that demonstrate ELMo's ability to implicitly learn logic rules. Finally, a crowdsourced analysis reveals how ELMo outperforms baseline models even on sentences with ambiguous sentiment labels.

1 Introduction

In this paper, we explore the effectiveness of methods designed to improve sentiment classification (positive vs. negative) of sentences that contain complex syntactic structures. While simple bag-of-words or lexicon-based methods (Pang and Lee, 2005; Wang and Manning, 2012; Iyyer et al., 2015) achieve good performance on this task, they are unequipped to deal with syntactic structures that affect sentiment, such as contrastive conjunctions (i.e., sentences of the form *"A-but-B"*) or negations. Neural models that explicitly encode word order (Kim, 2014), syntax (Socher et al., 2013; Tai et al., 2015) and semantic features (Li et al., 2017) have been proposed with the aim of improving performance on these more complicated sentences. Recently, Hu et al. (2016) incorporate logical rules into a neural model and show that these rules increase the model's accuracy on sentences containing contrastive conjunctions, while Peters et al. (2018a) demonstrate increased overall accuracy on sentiment analysis by initializing a model with representations from a language model trained on millions of sentences.

In this work, we carry out an in-depth study of the effectiveness of the techniques in Hu et al. (2016) and Peters et al. (2018a) for sentiment classification of complex sentences. Part of our contribution is to identify an important gap in the methodology used in Hu et al. (2016) for performance measurement, which is addressed by averaging the experiments over several executions. With the averaging in place, we obtain three key findings: (1) the improvements in Hu et al. (2016) can almost entirely be attributed to just one of their two proposed mechanisms and are also less pronounced than previously reported; (2) contextualized word embeddings (Peters et al., 2018a) incorporate the "A-but-B" rules more effectively without explicitly programming for them; and (3) an analysis using crowdsourcing reveals a bigger picture where the errors in the automated systems have a striking correlation with the inherent sentiment-ambiguity in the data.

2 Logic Rules in Sentiment Classification

Here we briefly review background from Hu et al. (2016) to provide a foundation for our reanalysis in the next section. We focus on a logic rule for sentences containing an *"A-but-B"* structure (the only rule for which Hu et al. (2016) provide experimental results). Intuitively, the logic rule for such sentences is that the sentiment associated with the whole sentence should be the same as the sentiment associated with phrase *"B"*.[1]

[1] The rule is vacuously true if the sentence does not have this structure.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4743–4751
Brussels, Belgium, October 31 - November 4, 2018.

More formally, let $p_\theta(y|x)$ denote the probability assigned to the label $y \in \{+,-\}$ for an input x by the baseline model using parameters θ. A logic rule is (softly) encoded as a variable $r_\theta(x,y) \in [0,1]$ indicating how well labeling x with y satisfies the rule. For the case of *A-but-B* sentences, $r_\theta(x,y) = p_\theta(y|B)$ if x has the structure *A-but-B* (and 1 otherwise). Next, we discuss the two techniques from Hu et al. (2016) for incorporating rules into models: *projection*, which directly alters a trained model, and *distillation*, which progressively adjusts the loss function during training.

Projection. The first technique is to *project a trained model into a rule-regularized subspace*, in a fashion similar to Ganchev et al. (2010). More precisely, a given model p_θ is projected to a model q_θ defined by the optimum value of q in the following optimization problem:[2]

$$\min_{q,\xi \geq 0} \mathrm{KL}(q(X,Y)||p_\theta(X,Y)) + C \sum_{x \in X} \xi_x$$
$$\text{s.t. } (1 - \mathbb{E}_{y \leftarrow q(\cdot|x)}[r_\theta(x,y)]) \leq \xi_x$$

Here $q(X,Y)$ denotes the distribution of (x,y) when x is drawn uniformly from the set X and y is drawn according to $q(\cdot|x)$.

Iterative Rule Knowledge Distillation. The second technique is to transfer the domain knowledge encoded in the logic rules into a neural network's parameters. Following Hinton et al. (2015), a "student" model p_θ can learn from the "teacher" model q_θ, by using a loss function $\pi H(p_\theta, P_{\text{true}}) + (1-\pi)H(p_\theta, q_\theta)$ during training, where P_{true} denotes the distribution implied by the ground truth, $H(\cdot,\cdot)$ denotes the cross-entropy function, and π is a hyperparameter. Hu et al. (2016) computes q_θ after every gradient update by projecting the current p_θ, as described above. Note that both mechanisms can be combined: After fully training p_θ using the iterative distillation process above, the projection step can be applied one more time to obtain q_θ which is then used as the trained model.

Dataset. All of our experiments (as well as those in Hu et al. (2016)) use the SST2 dataset, a binarized subset of the popular Stanford Sentiment Treebank (SST) (Socher et al., 2013). The dataset includes phrase-level labels in addition to sentence-level labels (see Table 1 for detailed statistics); following Hu et al. (2016), we use both types of labels for the comparisons in Section 3.2. In all other experiments, we use only sentence-level labels, and our baseline model for all experiments is the CNN architecture from Kim (2014).

3 A Reanalysis

In this section we reanalyze the effectiveness of the techniques of Hu et al. (2016) and find that most of the performance gain is due to projection and not knowledge distillation. The discrepancy with the original analysis can be attributed to the relatively small dataset and the resulting variance across random initializations. We start by analyzing the baseline CNN by Kim (2014) to point out the need for an averaged analysis.

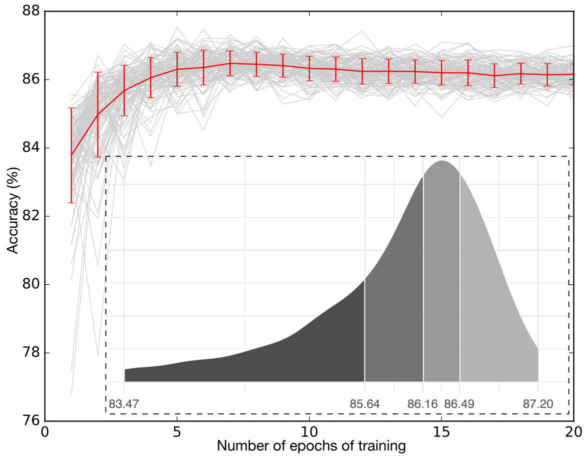

Figure 1: Variation in models trained on SST-2 (sentence-only). Accuracies of 100 randomly initialized models are plotted against the number of epochs of training (in gray), along with their average accuracies (in red, with 95% confidence interval error bars). The inset density plot shows the distribution of accuracies when trained with early stopping.

3.1 Importance of Averaging

We run the baseline CNN by Kim (2014) across 100 random seeds, training on sentence-level la-

Number of	Phrases	Train	Dev	Test
Instances	76961	6920	872	1821
A-but-B	3.5%	11.1%	11.5%	11.5%
Negations	2.0%	17.5%	18.3%	17.2%
Discourse	5.0%	24.6%	26.0%	24.5%

Table 1: Statistics of SST2 dataset. Here "Discourse" includes both *A-but-B* and negation sentences. The mean length of sentences is in terms of the word count.

[2] The formulation in Hu et al. (2016) includes another hyperparameter λ per rule, to control its relative importance; when there is only one rule, as in our case, this parameter can be absorbed into C.

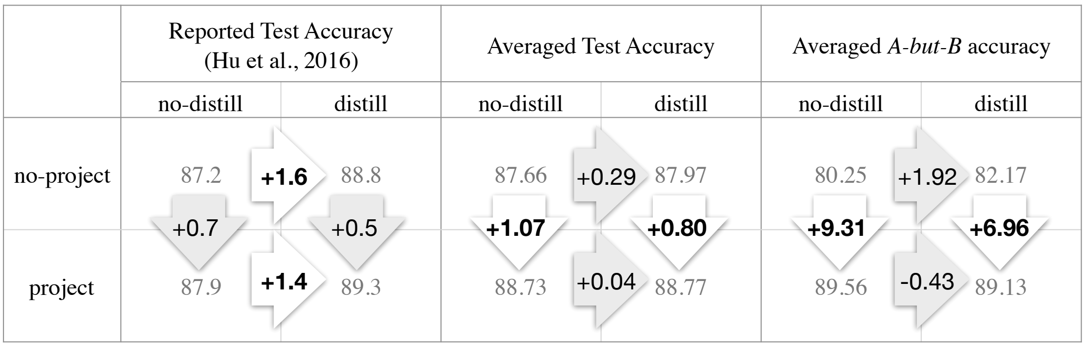

Figure 2: Comparison of the accuracy improvements reported in Hu et al. (2016) and those obtained by averaging over 100 random seeds. The last two columns show the (averaged) accuracy improvements for *A-but-B* style sentences. All models use the publicly available implementation of Hu et al. (2016) trained on phrase-level SST2 data.

bels. We observe a large amount of variation from run-to-run, which is unsurprising given the small dataset size. The inset density plot in Figure 1 shows the range of accuracies (83.47 to 87.20) along with 25, 50 and 75 percentiles.[3] The figure also shows how the variance persists even after the average converges: the accuracies of 100 models trained for 20 epochs each are plotted in gray, and their average is shown in red.

We conclude that, to be reproducible, only averaged accuracies should be reported in this task and dataset. This mirrors the conclusion from a detailed analysis by Reimers and Gurevych (2017) in the context of named entity recognition.

3.2 Performance of Hu et al. (2016)

We carry out an averaged analysis of the publicly available implementation[4] of Hu et al. (2016). Our analysis reveals that the reported performance of their two mechanisms (projection and distillation) is in fact affected by the high variability across random seeds. Our more robust averaged analysis yields a somewhat different conclusion of their effectiveness.

In Figure 2, the first two columns show the reported accuracies in Hu et al. (2016) for models trained with and without distillation (corresponding to using values $\pi = 1$ and $\pi = 0.95^t$ in the t^{th} epoch, respectively). The two rows show the results for models with and without a final projection into the rule-regularized space. We keep our hyper-parameters identical to Hu et al. (2016).[5]

The baseline system (no-project, no-distill) is identical to the system of Kim (2014). All the systems are trained on the phrase-level SST2 dataset with early stopping on the development set. The number inside each arrow indicates the improvement in accuracy by adding either the projection or the distillation component to the training algorithm. Note that the reported figures suggest that while both components help in improving accuracy, the distillation component is much *more* helpful than the projection component.

The next two columns, which show the results of repeating the above analysis after averaging over 100 random seeds, contradict this claim. The averaged figures show lower overall accuracy increases, and, more importantly, they attribute these improvements almost entirely to the projection component rather than the distillation component. To confirm this result, we repeat our averaged analysis restricted to only *"A-but-B"* sentences targeted by the rule (shown in the last two columns). We again observe that the effect of projection is pronounced, while distillation offers little or no advantage in comparison.

4 Contextualized Word Embeddings

Traditional context-independent word embeddings like `word2vec` (Mikolov et al., 2013) or `GloVe` (Pennington et al., 2014) are fixed vectors for every word in the vocabulary. In contrast, contextualized embeddings are dynamic representations, dependent on the current context of the word. We hypothesize that contextualized word embeddings might inherently capture these logic rules due to increasing the effective context size for the CNN layer in Kim (2014). Following the recent success of ELMo (Peters et al., 2018a) in sentiment analysis, we utilize the TensorFlow Hub implementation of ELMo[6] and feed these contextualized embeddings into our CNN model. We

[3] We use early stopping based on validation performance for all models in the density plot.

[4] `https://github.com/ZhitingHu/logicnn/`

[5] In particular, $C = 6$ for projection.

[6] `https://tfhub.dev/google/elmo/1`

fine-tune the ELMo LSTM weights along with the CNN weights on the downstream CNN task. As in Section 3, we check performance with and without the final projection into the rule-regularized space. We present our results in Table 2.

Switching to ELMo word embeddings improves performance by 2.9 percentage points on an average, corresponding to about 53 test sentences. Of these, about 32 sentences (60% of the improvement) correspond to *A-but-B* and negation style sentences, which is substantial when considering that only 24.5% of test sentences include these discourse relations (Table 1). As further evidence that ELMo helps on these specific constructions, the non-ELMo baseline model (no-project, no-distill) gets 255 sentences wrong in the test corpus on average, only 89 (34.8%) of which are *A-but-B* style or negations.

Statistical Significance: Using a two-sided Kolmogorov-Smirnov statistic (Massey Jr, 1951) with $\alpha = 0.001$ for the results in Table 2, we find that ELMo and projection each yield statistically significant improvements, but distillation does not. Also, with ELMo, projection is not significant. Specific comparisons have been added in the Appendix, in Table A3.

KL Divergence Analysis: We observe no significant gains by projecting a trained ELMo model into an *A-but-B* rule-regularized space, unlike the other models. We confirm that ELMo's predictions are much closer to the *A-but-B* rule's manifold than those of the other models by computing $\mathrm{KL}(q_\theta || p_\theta)$ where p_θ and q_θ are the original and projected distributions: Averaged across all *A-but-B* sentences and 100 seeds, this gives $0.27, 0.26$ and 0.13 for the Kim (2014), Hu et al. (2016) with distillation and ELMo systems respectively.

Intra-sentence Similarity: To understand the information captured by ELMo embeddings for *A-but-B* sentences, we measure the cosine similarity between the word vectors of every pair of words within the *A-but-B* sentence (Peters et al., 2018b). We compare the intra-sentence similarity for fine-tuned `word2vec` embeddings (baseline), ELMo embeddings without fine-tuning and finally fine-tuned ELMo embeddings in Figure 3. In the fine-tuned ELMo embeddings, we notice the words within the *A* and within the *B* part of the *A-but-B* sentence share the same part of the vector space. This pattern is less visible in the ELMo embeddings without fine-tuning and absent in the `word2vec` embeddings. This observation is indicative of ELMo's ability to learn specific rules for *A-but-B* sentences in sentiment classification. More intra-sentence similarity heatmaps for *A-but-B* sentences are in Figure A1.

Model		Test	*but*	*but* or *neg*
no-distill	no-project	85.98	78.69	80.13
no-distill	project	86.54	83.40	-
distill[7]	no-project	86.11	79.04	-
distill	project	86.62	83.32	-
ELMo	no-project	88.89	86.51	87.24
ELMo	project	88.96	87.20	-

Table 2: Average performance (across 100 seeds) of ELMo on the SST2 task. We show performance on *A-but-B* sentences ("*but*"), negations ("*neg*").

5 Crowdsourced Experiments

We conduct a crowdsourced analysis that reveals that SST2 data has significant levels of ambiguity even for human labelers. We discover that ELMo's performance improvements over the baseline are robust across varying levels of ambiguity, whereas the advantage of Hu et al. (2016) is reversed in sentences of low ambiguity (restricting to *A-but-B* style sentences).

Our crowdsourced experiment was conducted on Figure Eight.[8] *Nine* workers scored the sentiment of each *A-but-B* and negation sentence in the test SST2 split as 0 (negative), 0.5 (neutral) or 1 (positive). (SST originally had *three* crowdworkers choose a sentiment rating from 1 to 25 for every phrase.) More details regarding the crowd experiment's parameters have been provided in Appendix A.

We average the scores across all users for each sentence. Sentences with a score in the range $(x, 1]$ are marked as positive (where $x \in [0.5, 1)$), sentences in $[0, 1-x)$ marked as negative, and sentences in $[1-x, x]$ are marked as neutral. For instance, "*flat , but with a revelatory performance by michelle williams*" (score=0.56) is neutral when $x = 0.6$.[9] We present statistics of our dataset[10] in Table 3. Inter-annotator agree-

[7] Trained on sentences and not phrase-level labels for a fair comparison with baseline and ELMo, unlike Section 3.2.

[8] `https://www.figure-eight.com/`

[9] More examples of neutral sentences have been provided in the Appendix in Table A1, as well as a few "flipped" sentences receiving an average score opposite to their SST2 label (Table A2).

[10] The dataset along with source code can be found in

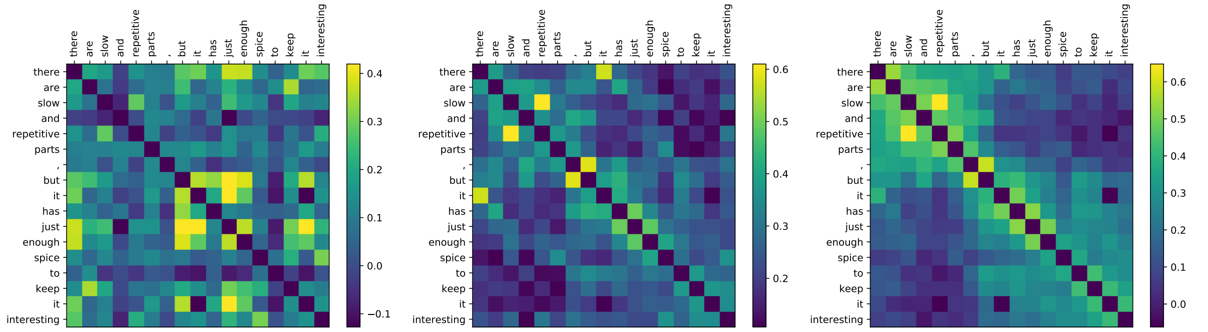

Figure 3: Heat map showing the cosine similarity between pairs of word vectors within a single sentence. The left figure has fine-tuned `word2vec` embeddings. The middle figure contains the original ELMo embeddings without any fine-tuning. The right figure contains fine-tuned ELMo embeddings. For better visualization, the cosine similarity between identical words has been set equal to the minimum value in the heat map.

ment was computed using Fleiss' Kappa (κ). As expected, inter-annotator agreement is higher for higher thresholds (less ambiguous sentences). According to Landis and Koch (1977), $\kappa \in (0.2, 0.4]$ corresponds to "fair agreement", whereas $\kappa \in (0.4, 0.6]$ corresponds to "moderate agreement".

We next compute the accuracy of our model for each threshold by removing the corresponding neutral sentences. Higher thresholds correspond to sets of less ambiguous sentences. Table 3 shows that ELMo's performance gains in Table 2 extends across all thresholds. In Figure 4 we compare all the models on the *A-but-B* sentences in this set. Across all thresholds, we notice trends similar to previous sections: 1) ELMo performs the best among all models on *A-but-B* style sentences, and projection results in only a slight improvement; 2) models in Hu et al. (2016) (with and without distillation) benefit considerably from projection; but 3) distillation offers little improvement (with or without projection). Also, as the ambiguity threshold increases, we see decreasing gains from projection on all models. In fact, beyond the 0.85 threshold, projection degrades the average performance, indicating that projection is useful for more ambiguous sentences.

Threshold	0.50	0.66	0.75	0.90
Neutral Sentiment	10	70	95	234
Flipped Sentiment	15	4	2	0
Fleiss' Kappa (κ)	0.38	0.42	0.44	0.58
no-distill, no-project	81.32	83.54	84.54	87.55
ELMo, no-project	87.56	90.00	91.31	93.14

Table 3: Number of sentences in the crowdsourced study (447 sentences) which got marked as neutral and which got the opposite of their labels in the SST2 dataset, using various thresholds. Inter-annotator agreement is computed using Fleiss' Kappa. Average accuracies of the baseline and ELMo (over 100 seeds) on non-neutral sentences are also shown.

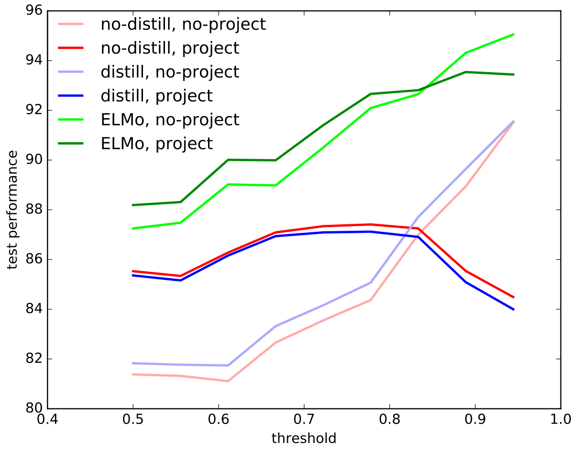

Figure 4: Average performance on the *A-but-B* part of the crowd-sourced dataset (210 sentences, 100 seeds)). For each threshold, only non-neutral sentences are used for evaluation.

6 Conclusion

We present an analysis comparing techniques for incorporating logic rules into sentiment classification systems. Our analysis included a meta-study highlighting the issue of stochasticity in performance across runs and the inherent ambiguity in the sentiment classification task itself, which was tackled using an averaged analysis and a crowdsourced experiment identifying ambiguous sentences. We present evidence that a recently proposed contextualized word embedding model (ELMo) (Peters et al., 2018a) implicitly learns logic rules for sentiment classification of complex sentences like *A-but-B* sentences. Future work includes a fine-grained quantitative study of ELMo word vectors for logically complex sentences along the lines of Peters et al. (2018b).

https://github.com/martiansideofthemoon/logic-rules-sentiment.

References

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop*.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. In *Association for Computational Linguistics (ACL)*.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Association for Computational Linguistics (ACL)*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Empirical Methods in Natural Language Processing (EMNLP)*.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

Shen Li, Zhe Zhao, Tao Liu, Renfen Hu, and Xiaoyong Du. 2017. Initializing convolutional filters with semantic features for text classification. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Frank J Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Association for Computational Linguistics (ACL)*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *North American Association for Computational Linguistics (NAACL)*.

Matthew E. Peters, Mark Neumann, Wen tau Yih, and Luke Zettlemoyer. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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 *Empirical Methods in Natural Language Processing (EMNLP)*.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Association for Computational Linguistics (ACL)*.

S. I. Wang and C. Manning. 2012. Baselines and bigrams: Simple, good sentiment and text classification. In *Association for Computational Linguistics (ACL)*.

Appendix

A Crowdsourcing Details

Crowd workers residing in five English-speaking countries (United States, United Kingdom, New Zealand, Australia and Canada) were hired. Each crowd worker had a Level 2 or higher rating on Figure Eight, which corresponds to a “group of more experienced, higher accuracy contributors”. Each contributor had to pass a test questionnaire to be eligible to take part in the experiment. Test questions were also hidden throughout the task and untrusted contributions were removed from the final dataset. For greater quality control, an upper limit of 75 judgments per contributor was enforced.

Crowd workers were paid a total of $1 for 50 judgments. An internal unpaid workforce (including the first and second author of the paper) of 7 contributors was used to speed up data collection.

# Judgments			Average	Sentence
Positive	Negative	Neutral		
1	1	7	0.50	the fight scenes are fun , but it grows tedious
3	2	4	0.56	it 's not exactly a gourmet meal but the fare is fair , even coming from the drive thru
2	3	4	0.44	propelled not by characters but by caricatures
4	2	3	0.61	not everything works , but the average is higher than in mary and most other recent comedies

Table A1: Examples of neutral sentences for a threshold of 0.66

# Judgments			Average	Original	Sentence
Positive	Negative	Neutral			
1	5	3	0.28	Positive	de niro and mcdormand give solid performances , but their screen time is sabotaged by the story 's inability to create interest
6	0	3	0.83	Negative	son of the bride may be a good half hour too long but comes replete with a flattering sense of mystery and quietness
0	5	4	0.22	Positive	wasabi is slight fare indeed , with the entire project having the feel of something tossed off quickly (like one of hubert 's punches) , but it should go down smoothly enough with popcorn

Table A2: Examples of flipped sentiment sentences, for a threshold of 0.66

Model 1		vs	Model 2		Significant
distill	no-project		distill	project	Yes
no-distill	no-project		no-distill	project	Yes
ELMo	no-project		ELMo	project	No
no-distill	no-project		distill	no-project	No
no-distill	project		distill	project	No
no-distill	no-project		ELMo	no-project	Yes
distill	no-project		ELMo	no-project	Yes
no-distill	project		ELMo	project	Yes
distill	project		ELMo	project	Yes

Table A3: Statistical significance using a two-sided Kolmogorov-Smirnov statistic (Massey Jr, 1951) with $\alpha = 0.001$.

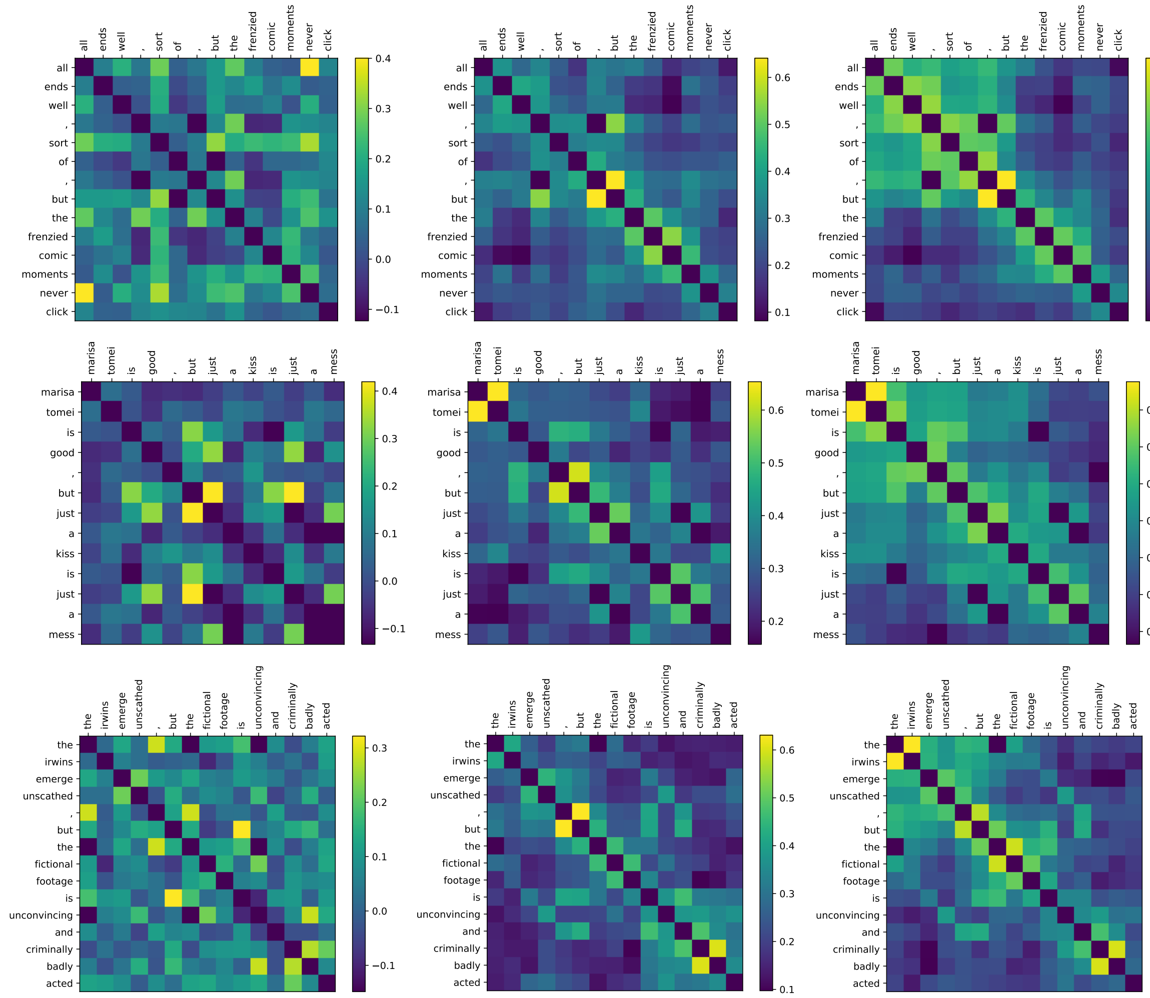

Figure A1: Heat map showing the cosine similarity between pairs of word vectors within a single sentence. The leftmost column has `word2vec` (Mikolov et al., 2013) embeddings, fine-tuned on the downstream task (SST2). The middle column contains the original ELMo embeddings (Peters et al., 2018a) without any fine-tuning. The representations from the three layers (token layer and two LSTM layers) have been averaged. The rightmost column contains ELMo embeddings fine-tuned on the downstream task. For better visualization, the cosine similarity between identical words has been set equal to the minimum value in the map.