An Evaluation Dataset and Strategy for Building Robust Multi-turn Response Selection Model

Kijong Han[1]*, Seojin Lee[2*†], Wooin Lee[1], Joosung Lee[1], and Dong-hun Lee[1‡]

[1]Kakao Enterprise, South Korea
[2]SK Telecom, South Korea
{mat.h,dan.kes,rung.joo,hubert.std}@kakaoenterprise.com
seojin.lee@sktair.com

Abstract

Multi-turn response selection models have recently shown comparable performance to humans in several benchmark datasets. However, in the real environment, these models often have weaknesses, such as making incorrect predictions based heavily on superficial patterns without a comprehensive understanding of the context. For example, these models often give a high score to the wrong response candidate containing several keywords related to the context but using the inconsistent tense. In this study, we analyze the weaknesses of the open-domain Korean Multi-turn response selection models and publish an adversarial dataset to evaluate these weaknesses. We also suggest a strategy to build a robust model in this adversarial environment.

1 Introduction

Multi-turn response selection is a task that selects the best response among given candidates for a given dialogue context. Response selection models have recently shown comparable performance to humans (Cui et al., 2020) in the several in-domain/held-out benchmarks (Lowe et al., 2015; Zhang et al., 2018a; Dinan et al., 2020). However, in the actual service environment, these models are often found to have weaknesses. For example, the model gives the highest score to the wrong response, which has high word overlap with the context (Yuan et al., 2019) or semantically similar to the context (Whang et al., 2021).

Held-out evaluation often overestimates the real-world performance of the model (Ribeiro et al., 2020), so adversarial datasets for evaluating weaknesses have been constructed for each task, such as NLI (Naik et al., 2018; McCoy et al., 2019), and MRC (Jia and Liang, 2017; Rajpurkar et al., 2018). A framework for comprehensively evaluating the general linguistic abilities of the model was also studied (Ribeiro et al., 2020).

Several works evaluated adversarial cases for the response selection task (Yuan et al., 2019; Whang et al., 2021). However, they just automatically generate adversarial responses by copying words in the context. In this study, we analyze the weaknesses of the various aspects of the open-domain Korean Multi-turn Response Selection models and construct an adversarial dataset manually. A total of 2,220 test cases are constructed, and each test case are classified by type.

Neural networks do not generalize well to such an adversarial setting because they tend to use superficial patterns and spurious correlation of the dataset overly, which makes models biased(Clark et al., 2019; Nam et al., 2020). Thus, various debiasing methods have been studied to alleviate this phenomenon (He et al., 2019; Utama et al., 2020). In this study, we show that debasing method is also effective in adeversarial evaluation for multi-turn response selection task.

In the retrieval-based chatbot system where response selection is used, response candidates are composed as follows. All utterances in the database are used as response candidates (Humeau et al., 2020), or part of them filtered through search engines are used (Zhou et al., 2020). To filter the candidates, machine learning-based embeddings or word-level similarity algorithms(e.g., BM25), which also have weaknesses in an adversarial setting, are used (Zhou et al., 2020). Therefore, almost every time a response is selected by the actual system, adversarial cases are included in the candidates. Thus, robustness to adversarial cases is more important for response selection task. We also construct a real environment test set and experiment that the model robust to an adversarial case has high performance in the real environment.

*These authors contributed equally to this work.
† This work was done while the author was working at Kakao Enterprise.
‡ Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2338–2344
November 7–11, 2021.

Type	Context	Adversarial Response	# cases
Repetition	[A] I'm hungry / [B] What do you want to eat?	I'm hungry	400
Negation	[A] Wrap up before you go outside / [B] Why? [A] It's freezing cold.	Yes, indeed. it's **not** that cold today.	454
Tense	[A] I can't wait to watch "Joker" / [B] I watched the movie. It was really impressive. / [A] Wow! I should watch it.	You really enjoy**ed** it.	158
Subject-Object	[A] I'm in love with BTS / [B] Why do you like them so much? / [A] ϕ(their) Songs are great	Thanks (for complimenting me)	374
Lexcial Contradiction	[A] It's freezing **cold** today.	Yes, indeed. It's way too **hot** out today.	254
Interrogative Word	[A] I saw Jennie today / [B] What does she look like? / [A] ϕ(she) Looks so pretty	**Who**'s so pretty?	236
Topic	[A] Isn't the **weather** nice today? / [B] Oh, is it? [A] Yeah, it's sunny and warm.	Bring your **umbrella** with you.	344

Table 1: Examples of adversarial data for each type. ϕ denotes a zero anaphora in Korean.

2 Adversarial Test Dataset

We analyze the incorrect responses in the internal service log and categorize the types of frequent errors. There are a total of seven types, and details of each type are as follows.

Repetition An incorrect response repeating one of the utterances in the context.

Negation A negation is either added to or omitted from a correct response, generating an erroneous response with reversed affirmative or negative meaning. A test set for a negation error intentionally generates a negative response by adding or removing '안' or '못', which are negative adverbs in Korean (short-form negation) or '-지 않다,' '-지 못하다, ' or '-지 말다' which are negative auxiliary predicates in Korean (Long-form negation) in order to test whether the model understands such semantic reversal.

Tense A morpheme or expression marking tense is added to or removed from a correct response, generating an erroneous response in tense that is inconsistent with the given context. A test set for tense errors adds or replaces morphemes or expressions marking the future tense such as '-겠-,' or ones marking the past tense such as'-었-' to test whether the model fully understands the context disconnection triggered by such tense change.

Subject-Object A test set for subject-object errors generates a response inconsistent with the context due to confusion of the subject and object for a certain action. In particular, since zero anaphora can be found frequently in Korean sentences, incorrect responses are often made because of a failure in identifying the hidden subject of the previous context. This test set uses a subject or an object differently from the ones used in a correct response to examine whether the model fully understands the context disconnection caused by such errors.

Lexical Contradiction A key lexicon of a correct response is replaced with one that holds either conflicting or opposite meaning against the said key lexicon, generating an incorrect response. A test set for lexical contradiction errors replaces a key lexicon in a sentence with an antonym (e.g. hot vs cold) or a word that cannot be used instead (e.g. rain vs snow) to check whether the model understands the precise meaning of such lexicon.

Interrogative Word A test set for interrogative word errors generates a response in a form of 5W1H questions to ask for information that has already been explicitly or implicitly shared in previous dialogues.

Topic A key sentence or vocabulary is replaced with another sentence or term that does not fit in the previous context even though they frequently appear together in the given topic. While this error is similar to the lexical contradiction error to a certain extent, the replacement words used in this test do not hold conflicting or opposite meanings but instead have less semantic relevance to the context of the previous dialogue (e.g. sunny vs umbrella). The test set assesses whether a model fully understands the fact that while the replacement vocabulary is the one that is frequently used in the same given topic, the response does not correctly reflect the context of the previous dialogue.

Five annotators generate a total of 200 dialogue sessions. For each session i, annotators create two correct responses and an arbitrary number(M_i) of incorrect responses based on the instruction described above. All sessions and responses are reviewed and filtered by experts. We set up one test case to consist of context, one correct response, and one incorrect response. Therefore, $2 * M_i$ test

cases were extracted for each session, and a total of 2,220 test cases are constructed. It evaluates whether the model gives the correct answer a higher score than the incorrect one for a given context. Statistics and examples are described in Table 1. We release this data set at https://github.com/kakaoenterprise/KorAdvMRSTestData.

3 Method

Suppose that dataset is $D = \{(c_i, r_i, y_i)\}_{i=1}^{N}$, where c_i denotes a dialogue context, r_i is a response utterance, and $y_i \in \{0, 1\}$ is a label. The context $c_i = \{u_{i,1}, u_{i,2}, ..., u_{i,k_i}\}$ consists of sequence of k_i utterances. The label $y_i = 1$ means that r_i is sensible response for context c_i.

3.1 Baseline: Fine-tuning BERT

We adopt fine-tuning BERT (Devlin et al., 2019) as a baseline. In this work, similar to the previous works that fine-tuned BERT for the Multi-turn Response Selection task (Gu et al., 2020; Whang et al., 2020; Han et al., 2021), the input token sequence of BERT x_i is composed as follows.

$$x_i = [CLS]u_{i,1}[EOU]...u_{i,k_i}[EOU][SEP] \\ r_i[EOU][SEP] \quad (1)$$

The [EOU] is a special token indicating that the utterance is over. The final output hidden vector of the [CLS] token in BERT is fed into a fully connected layer with softmax activation. Then, the BERT is fine-tuned to minimize cross entropy loss between the target label and output of this layer.

3.2 Debiasing Strategy

In general, correct dialogue response utilizes keywords or topics in the context. Neural networks tend to use such superficial patterns(e.g., keyword, topic) overly, which makes models biased (Clark et al., 2019; Nam et al., 2020). We see this bias as the main cause of the response selection model's vulnerability to an adversarial environment. Thus, we experimented by applying various debiasing techniques to the response selection task, and DRiFt (He et al., 2019) was the most effective. The main concept of the debiasing strategy we used is to train a debiased model to fit the residual of the biased model, focusing on examples that cannot be predicted well by biased features only (He et al., 2019). Details of the method using DRiFt are as follows.

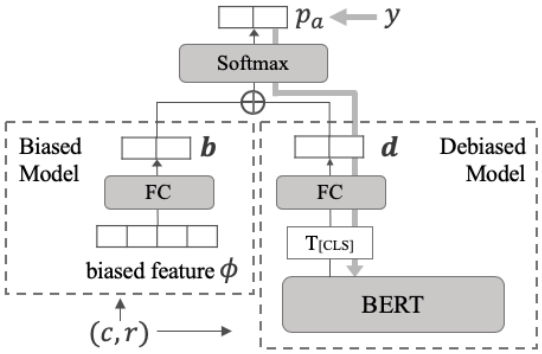

Figure 1: Overall architecture for training debiased model utilizing biased model. The grey line represents that gradient is backpropagated only to the debiased model.

First, we train an auxiliary biased model using only biased features. The biased model is a single fully connected layer with softmax activation and trained with cross-entropy loss. The biased feature vector used as an input ϕ_i is as follows.

$$\phi_i = [JS_{morph}(c_i, r_i), JS_{morph}(u_{i,k_i}, r_i), \\ JS_{wordpiece}(c_i, r_i), JS_{wordpiece}(u_{i,k_i}, r_i)] \quad (2)$$

We use the Jaccard similarity(JS) between the whole context(c_i) and response(r_i) as input features. We also use the JS between the last utterance(u_{i,k_i}) and r_i, because the last utterance is most important (Zhang et al., 2018b; Ma et al., 2019). We use two tokenizers: the WordPiece (Wu et al., 2016), and the morpheme analyzer. We assume that these words overlap feature could capture keyword and topic bias.

Second, we train a debiased model utilizing a biased model, as shown in Figure 1. The overall structure of the debiased model is the same as the baseline, but only the learning scheme is different. Let **b** is output hidden vector of the biased model, **d** is output hidden vector of the debiased model, $p_b = softmax(\mathbf{b})$, and $p_d = softmax(\mathbf{d})$. DRiFt method minimize cross entropy loss between $p_a = softmax(\mathbf{b} + \mathbf{d})$ and target labels. Thus, the loss function is defined as follows.

$$Loss = -\log p_a(y_i) = \\ -\log p_b(y_i) - \log p_d(y_i) + \log \sum_{l=0}^{L-1} p_b(l) p_d(l) \quad (3)$$

L is the number of classification classes(2 for this task). The gradient is backpropagated only to the

	Train	Val	In-domain Test	Adv Test	Real Test
# pairs	500K	10K	10K	4,440	5,490
# cands	2	10	10	2	10
pos:neg	1:1	1:9	1:9	1:1	4.4:5.6
# turns	4.6	4.5	4.6	2.9	3.3

Table 2: Statistics for each dataset.

Method	In-domain	Adversarial	Real Env.
baseline	86.4±0.5	39.4±1.7	36.2±0.9
+deb	85.4±0.7	43.5±2.3	36.5±0.6
+UMS	**87.5±0.7**	42.9±2.5	38.8±0.6
+UMS+deb	87.0±0.7	**47.2±2.1**	**40.4±0.7**

Table 3: Overall performance of each method.

Type	Repetition	Negation	Tense	Subject -Object	Lexical Contradiction	Interrogative Word	Topic
baseline	12.9±2.5	36.1±2.3	41.8±2.3	55.0±1.9	41.1±1.8	46.1±2.2	50.7±1.8
+deb	26.2±5.1	**40.6±2.8**	43.0±2.6	56.2±1.9	**43.9±1.6**	45.5±3.2	51.8±1.9
+UMS	26.2±6.2	34.5±1.8	45.3±2.0	61.2±2.9	41.6±3.1	**46.4±2.0**	51.1±1.6
+UMS+deb	**40.0±5.4**	38.5±1.1	**46.9±2.3**	**63.9±3.0**	43.3±1.7	45.9±2.6	**52.8±1.7**

Table 4: Performance for each adversarial type.

debiased model. The last term encourages output from the debiased model p_d, to have minimal projection on output from the biased model p_b (He et al., 2019). Derivation of equation 3 is in Appendix A. At test time, only debiased model is used.

3.3 Combination with Multi-task Learning

Recently, self-supervised learning approaches have shown state-of-the-art performance in the response selection task (Whang et al., 2021; Xu et al., 2021). These works devise auxiliary tasks to understand the dialogue better and train the model in a multi-task manner. The final loss function in these methods is the weighted sum of losses of auxiliary tasks and main task (i.e., determine given response is a sensible response to the context). Thus, debiasing strategy could be easily combined with these methods by replacing the loss function of the main task with equation 3. We also experiment with self-supervised learning approach UMS (Whang et al., 2021), and we show that it is also effective in not only in-domain but also adversarial and real environments.

4 Experiments and Results

4.1 Experiment Setup

We construct an experimental dataset using the corpus that we produced in-house and the public Korean dialogue corpus[1]. We split these corpora into three, and each is for training, validation, and test. Statistics of each dataset are described in Table 2. #pairs denote the number of context-response pairs, #cands denotes the number of candidates per context, pos:neg denotes the ratio of positive and negative responses in candidates, and #turns denote the average turns per context. Details on the construction are as follows.

Train, valid, and in-domain test The last utterance of the dialogue session is used as a positive response and the rest as context. Negative responses are randomly chosen from the other dialogue.

Adversarial test It is described in the Section 2.

Real environment test In a real environment, response candidates are not sampled randomly but are sampled through a search system (Zhou et al., 2020), or all utterances without sampling are used as candidates (Humeau et al., 2020). There are many adversarial negatives in this situation, as described in Section 1. We build a dataset by simulating this situation in a similar way to the previous works (Wu et al., 2017; Zhang et al., 2018b).

We take a dialogue session from the test corpus and internal service log as context. We trained a bi-encoder-based context and response embedding model (Humeau et al., 2020) and indexed embeddings of all utterances in the corpus. Then, we retrieve the top 10 utterances based on the similarity score between context embedding as response candidates. For each response, three annotators labeled whether it is sensible to the context. The response determined by more than two people as sensible was selected as the positive response.

4.2 Results

We measure the performance ten times for each model and report the mean and standard deviation in Table 3. See Appendix B for details of training. The baseline is a fine-tuned BERT described in Section 3.1. "deb" denotes a debiasing strategy described in Section 3.2. UMS denotes a self-supervised multi-task learning method de-

[1] https://corpus.korean.go.kr

scribed in Section 3.3. Precision@1 is used as an evaluation metric for all test sets.

Debiasing strategy significantly improves adversarial test performance in both baseline and UMS model; it achieves absolute improvements of 4.1% and 4.3% on baseline and UMS. A decline in performance is observed in the in-domain test; -1.0% and -0.5% on baseline and UMS, as the DRiFt debiasing method (He et al., 2019) shows a slight performance degradation in the in-domain test. However, It improves performance in the comprehensive real environment test; +0.3% and +1.6% on baseline and UMS. This supports our argument that robustness to adversarial cases is important in the response selection task. Additionally, +UMS+deb outperforms +deb in all test set. From this, it can be seen that the debiasing strategy and UMS have a synergistic effect.

The performance of each adversarial type is reported in Table 4. Since we used word-level Jacard Similarity as a biased feature, the debiasing strategy shows huge performance improvement in the Repetition type, which simply uses word sequence in context as a negative response. There is no improvement in the Interrogative Word type. We assume that the reason for it is that this type is difficult because it requires understanding all 5W1H from the context.

5 Conclusion

We analyze the weaknesses of the open-domain Korean Multi-turn Response Selection models and publish an adversarial dataset to evaluate these weaknesses. We suggest a strategy to build a robust model to an adversarial and real environment with the experimental results. We expect that this work and dataset will help improve the response selection model.

6 Ethical Considerations

The adversarial dataset we publish is generated manually. All sessions and responses in the dataset are reviewed and filtered by the experts, and we also considered ethical issues in this process. Thus, there is no hate speech or privacy issue in our dataset.

References

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4060–4073.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. Mutual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1406–1416.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2020. The second conversational intelligence challenge (convai2). In *The NeurIPS'18 Competition*, pages 187–208. Springer.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.

Janghoon Han, Taesuk Hong, Byoungjae Kim, Youngjoong Ko, and Jungyun Seo. 2021. Fine-grained post-training for improving retrieval-based dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1549–1558.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. *EMNLP-IJCNLP 2019*, page 132.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Wentao Ma, Yiming Cui, Nan Shao, Su He, Weinan Zhang, Ting Liu, Shijin Wang, and Guoping Hu. 2019. Triplenet: Triple attention network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 737–746.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. *arXiv preprint arXiv:1806.00692*.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. 2020. Learning from failure: Training debiased classifier from biased classifier. *NeurIPS*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Mind the trade-off: Debiasing nlu models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729.

Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and Heuiseok Lim. 2020. An effective domain adaptive post-training method for bert in response selection. *Proc. Interspeech 2020*, pages 1585–1589.

Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. 2021. Do response selection models really know what's next? utterance manipulation strategies for multi-turn response selection. *AAAI*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505.

Ruijian Xu, Chongyang Tao, Daxin Jiang, Xueliang Zhao, Dongyan Zhao, and Rui Yan. 2021. Learning an effective context-response matching model with self-supervised tasks for retrieval-based dialogues. *AAAI*.

Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 111–120.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018b. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3740–3752.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93.

A Derivation of Loss Function

Let b is output hidden vector of the biased model, d is output hidden vector of the debiased model, $y_i \in \{0, 1\}$ is the label value, $p_b = softmax(b)$, $p_d = softmax(d)$, and $p_a = softmax(b + d)$.

$$
\begin{aligned}
Loss &= -\log p_a(y) \\
&= -\log e^{b_y + d_y} + \log \sum_l e^{b_l + d_l} \\
&= -\log e^{b_y} - \log e^{d_y} + \log \sum_l e^{b_l} e^{d_l} \\
&= -\log e^{b_y} - \log e^{d_y} + \log \sum_l e^{b_l} e^{d_l} \\
&\quad + \log \sum_l e^{b_l} - \log \sum_l e^{b_l} \\
&\quad + \log \sum_l e^{d_l} - \log \sum_l e^{d_l} \\
&= -(\log e^{b_y} - \log \sum_l e^{b_l}) \\
&\quad -(\log e^{d_y} - \log \sum_l e^{d_l}) \\
&\quad +(\log \sum_l e^{b_l} e^{d_l} - \log \sum_l e^{b_l} \sum_l e^{d_l}) \\
&= -\log \frac{e^{b_y}}{\sum_l e^{b_l}} - \log \frac{e^{d_y}}{\sum_l e^{d_l}} \\
&\quad + \log \sum_l \frac{e^{b_l} e^{d_l}}{\sum_l e^{b_l} \sum_l e^{d_l}} \\
&= -\log p_b(y) - \log p_d(y) + log \sum_l p_b(l) p_d(l)
\end{aligned}
$$

B Training Details

The biased model, which consists of a single fully connected layer, is trained using the AdamW optimizer with a learning rate of 5e-4 and for 3 epochs. BERT-based models, including baseline, UMS, and debiased models, are trained using the AdamW optimizer with a learning rate of 2.5e-5 and for 3 epochs on 4 Nvidia Volta v100 GPU. The batch size is 128 for every model. We train and evaluate 10 times for each model and calculate mean and standard deviation. For each model, a checkpoint that shows the best performance in the real environment is selected for performance measure.