Similarity Based Label Smoothing For Dialogue Generation

Sougata Saha, Souvik Das, Rohini Srihari
State University of New York at Buffalo
Department of Computer Science and Engineering
{sougatas, souvikda, rohini}@buffalo.edu

Abstract

Generative neural conversational systems are typically trained by minimizing the entropy loss between the training "hard" targets and the predicted logits. Performance gains and improved generalization are often achieved by employing regularization techniques like label smoothing, which converts the training "hard" targets to soft targets. However, label smoothing enforces a data-independent uniform distribution on the incorrect training targets, leading to a false assumption of equiprobability. In this paper, we propose and experiment with incorporating data-dependent word similarity-based weighing methods to transform the uniform distribution of the incorrect target probabilities in label smoothing to a more realistic distribution based on semantics. We introduce hyperparameters to control the incorrect target distribution and report significant performance gains over networks trained using standard label smoothing-based loss on two standard open-domain dialogue corpora.

1 Introduction

Response generators rely heavily on language modelling for response generation. Given a context comprising multiple conversation utterances, a response generator is formulated as a next utterance prediction problem, where the task is to generate a response conditioned on the context. With the advent of deep learning and availability of sufficient training data, parametric models like recurrent neural networks and transformers are generally used for language modelling. Trained by minimizing the expected cross entropy between the training targets and the prediction logits, such models often overfit the training data and does not generalize well on the test set. Label smoothing proposed by Szegedy et al. (2015) to improve the performance of Inception net image classifier on the ImageNet dataset has gained wide acceptance in Natural Language Processing tasks as a regularization technique to enhance the generalization capability of deep neural networks. Vaswani et al. (2017) in his work "Attention is all you need", where he proposed the state-of-the-art transformer architecture, had reported performance gains in machine translation using label smoothing during training. Unlike other regularization techniques which constrain the model parameters and hidden representations, label smoothing augments the actual targets by reducing the target probability and assigning low probabilities to all classes, following a data independent uniform distribution. Thus, preventing the model from predicting the correct labels overconfidently during training. However, as pointed out by Pereyra et al. (2017) and Hinton et al. (2015), the probabilities assigned to both the correct and incorrect classes constitute the knowledge of a network. In language modelling, incorporating label smoothing and assigning a uniform probability to all the incorrect classes can convey a false knowledge to the model. For example, as depicted in Figure 1, while generating "I am doing good" in response to the query "How are you doing ?", having generated the partial phrase "I am doing", both "great" and "awesome" conveys the same message as "good". Although "bad" would convey an opposite yet meaningful message, a random word like "aeroplane" would be inappropriate. Hence, instead of assuming a uniform distribution for the incorrect classes while using label smoothing, we can incorporate a weighing mechanism to present such knowledge to the

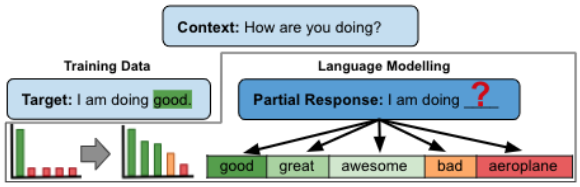

Figure 1: Sample conversation depicting token probability distribution using label smoothing, in comparison to desired distribution.

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 253 - 259

model. Here, we introduce simple ways of imparting such information by modifying the data independent uniform distribution in label smoothing with a more appropriate data dependent distribution proportional to the pre-trained word-embedding based similarity between the actual and incorrect targets. Our primary contributions are follows (i) We propose a robust mechanism for augmenting the target labels in language modelling, which better reflects the real-world. (ii) We experiment our proposed framework with different hyper-parameter settings and perform thorough analysis of the observations [1]

2 Related Work

Although numerous techniques have been introduced to enhance the generalizability of neural networks, as pointed out by Pereyra et al. (2017), most work focus on regularizing model parameters, compared to external regularization techniques like label smoothing or target data augmentation. Recent approaches for generalizable conversations can be broadly categorized as follows

Loss function augmentation: Li et al. (2016) proposed using Maximum Mutual Information along with the Cross Entropy loss, in order to penalize generic responses like "I do not know", which are frequent in conversational datasets. Jiang et al. (2019) attributed generic responses to the cross entropy loss function, which prefers frequent tokens. They proposed augmenting the loss function with a frequency based weighing mechanism dependent on the corpus for engendering diverse responses. Wang et al. (2020) experimented with using optimal transport to match sequences generated in the teacher and student modes, and increasing performance of student forced networks on the test dataset by reducing the gap between the two modes. Wang et al. (2021) proposed an adaptive label smoothing approach that can adaptively estimate a target label distribution at each time step for different contexts. Compared to their approach, our proposed method is simpler with fewer parameters.

Data augmentation: Cai et al. (2020) demonstrated that conversational datasets generally don't exhibit coherence in query response pairs, which affect the Cross Entropy loss. They propose a training data augmentation module, which can not only replace words in the actual target response with similar words using BERT (Devlin et al., 2019), but also augment the style of the response, preserving the meaning. They further introduced a neural weighting mechanism, which can assign weights or importance to the augmented and golden training data, and report significant performance gains. Kang and Hashimoto (2020) demonstrated that the log loss is not robust to noise, and hence proposed truncating the distribution of the training targets to achieve an easy to optimize and more robust loss function. He and Glass (2020) introduced a network which can provide negative generated samples, and train the generation model to maximize the log likelihood of training data while minimizing the likelihood of negative samples. Since instead of augmenting the training data, we adjust the probability of incorrect labels for each correct label, our proposed method belongs to the first category.

3 Methods and Experiments

We experiment with ways to augment the data independent uniform distribution enforced by label smoothing. Let U_i be an utterance consisting of words $\{w_j\}_{j=1}^{N}$, where N is the number of words in the utterance. For each word w_j, in label smoothing a probability $1-s$ is assigned to the true label w_j, and a probability of s (smoothing factor) is distributed uniformly among the rest of the k words in the vocabulary. We augment the distribution of the incorrect class by weighting the smoothing factor s according to the cosine similarity between the Glove (Pennington et al., 2014) word embedding of the correct word in the training data and all the words in the vocabulary. Thus, if the correct word to be predicted is "good", then the words "great" and "awesome" in the vocabulary would get a higher proportion of the smoothing factor s, compared to an unrelated word like "aeroplane"-presenting more accurate knowledge to the model. Mathematically, let $\vec{w_j}$ be the Glove word embedding of word w_j, $\mathbf{W_k}$ be a matrix containing the Glove word embedding for all the words in the vocabulary (including w_j), $\vec{w}_{j\_sim}$ be the vector of cosine similarity between the word w_j and all the words in the vocabulary. Since Glove word embeddings are learned representations, they can be noisy. Hence, we introduce a binary mask $mask_j$ using a threshold t, below which we set the cosine similarity value in $\vec{w}_{j\_sim}$ as 0, and multiply the similarity vector with the mask. The resulting vector is normalised to lie between 0 and 1, and finally multiplied by s. We treat t as a model hy-

[1] Code and dataset available here.

perparameter, and is tuned using grid search. We further reason that although Glove embeddings are learned from text corpora, there are possibilities that dissimilar words can lie in close proximity in the embedding space, resulting in a high cosine similarity score, and presenting an incorrect knowledge to the model. To circumvent this problem, we further experiment with filtering out the cosine similarities of dissimilar words based on WordNet sysnets (Miller, 1995), which we achieve by implementing another mask $mask_j^{'}$.

$$\vec{w}_{j\_sim} = \frac{\vec{w_j} \cdot \mathbf{W_k}}{||\vec{w_j}|| \cdot ||\mathbf{W_k}||} \tag{1}$$

$$\vec{w}_{j\_dist} = \frac{\vec{w}_{j\_sim} * mask_j * mask_j^{'}}{\sum(\vec{w}_{j\_sim} * mask_j * mask_j^{'})} * s \tag{2}$$

$$\vec{w}_{j\_dist}[j^*] = 1 - s \tag{3}$$

$$\text{where, } mask_j = \begin{cases} 0, & \text{if } \vec{w}_{j\_sim} <= t \\ 0, & \text{if } \vec{w}_{j\_sim} = 1 \\ 1, & \text{otherwise} \end{cases}$$

$$\text{and, } mask_j^{'} = \begin{cases} 1, & \text{if } w_k \text{ is a synonym of } w_j \\ 0, & \text{otherwise} \end{cases}$$

3.1 Dataset

We experiment with (i) The DailyDialog dataset (Li et al., 2017): A multi-turn open domain dialogue dataset comprising 13,118 conversations pertaining to diverse day-to-day topics, and (ii) The Empathetic Dialogues dataset (Rashkin et al., 2019): An open domain multi-turn dataset consisting of 25,000 conversations grounded in emotional situations. We use the same training, validation and testing splits as mentioned in the datasets. We concatenate all the turns in the query in one long text, and use two special tokens: *[speaker1]* and *[speaker2]* to distinguish the speakers. In order to speed up computation, we restrict the context to the most recent 50 tokens, which is determined analytically from the corpora. Additional training details and code in Section A.2 (Appendix A).

3.2 Model

Since the primary scope of this paper is to experiment with different loss functions, we used a standard transformer encoder-decoder architecture as proposed by Vaswani et al. (2017), where the encoder encodes the most recent utterance in the conversation, along with context from the previous turns. The encoder-decoder comprises of 3 layers each, with 300 dimensional hidden representation, with 6 attention heads in each multi-headed attention layer. The embedding layer is populated with 300 dimensional Glove embeddings, which are trained along with the entire network. Finally, a fully connected linear layer predicts the next word.

3.3 Experiments

We treat the Cross Entropy (CE) loss, CE loss with label smoothing, Kullback–Leibler (KL) divergence loss and KL loss with label smoothing as baselines. We experiment with different smoothing values $s \in \{\text{NA}, 0.1, 0.2\}$, cosine similarity thresholds $t \in \{\text{NA}, 0.0, 0.5, 0.8\}$, and also perform ablation study to analyze the usefulness of the WordNet similarity mask $w \in \{\text{NA}, 0, 1\}$. Overall we experiment with 30 diverse settings per dataset.

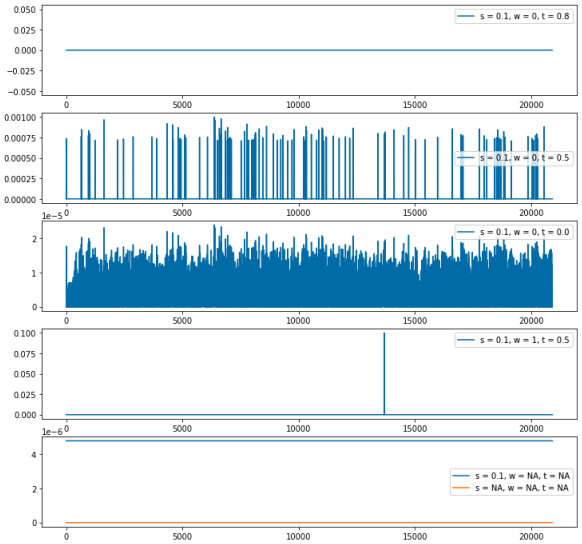

Figure 2: Illustration of incorrect word probabilities(x-axis = vocabulary, y-axis = probability). Setting t = 0.8, or using WordNet mask filters out most words making the target distribution equivalent to vanilla CE loss targets. Using t = 0.5 or 0.0 yields a less dramatic effect and preserves the information of the incorrect labels.

4 Results and Analysis

We compare the (i) sacreBLEU score (Post, 2018): a standardised version of the BLEU score (Papineni et al., 2002), (ii) ROUGE L score (Lin, 2004): which compares Longest Common Subsequence (LCS), and automatically takes into account sentence level structure similarity and identifies longest co-occurring in sequence n-grams, (iii) METEOR score (Banerjee and Lavie, 2005): an improvement over BLEU score, which incorporates stemming and synonymy matching along

			s = NA	s = 0.1	s = 0.2	s = 0.1				s = 0.2			
			t = NA	t = NA	t = NA	t = 0		t = 0.5		t = 0		t = 0.5	
Dataset	**Metric**	**Loss**	**w = NA**	**w = NA**	**w = NA**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**
DD	SacreBLEU	CE	1.662	1.852	1.725	1.989	1.762	**2.193 (+ 12.67%)**	1.857	2.115	1.802	2.053	1.930
		KL	**1.946**	1.753	1.793	1.845	1.818	1.912	1.885	1.938	1.809	1.729	1.885
	ROUGE L	CE	0.120	0.124	0.124	0.123	0.120	**0.127 (+ 0.57%)**	0.126	0.121	0.120	0.123	0.124
		KL	**0.126**	0.122	0.123	0.122	0.126	0.124	0.124	0.122	0.125	0.123	0.123
	METEOR	CE	0.124	**0.132**	0.128	0.134	0.128	**0.137 (+ 4.16%)**	0.131	0.134	0.127	0.134	0.131
		KL	**0.132**	0.130	0.130	0.134	0.132	0.132	0.131	0.131	0.131	0.129	0.129
ED	SacreBLEU	CE	2.279	**2.408**	2.190	**2.442 (+ 1.42%)**	2.208	2.192	2.216	2.318	2.262	2.312	2.256
		KL	2.271	2.168	2.279	2.277	2.278	2.337	2.361	2.431	2.274	2.439	2.143
	ROUGE L	CE	0.138	**0.143**	0.137	0.144	0.140	0.142	0.138	0.141	0.139	0.141	0.138
		KL	0.140	0.139	0.142	0.144	0.145	0.143	0.138	**0.146 (+ 1.95%)**	0.140	0.143	0.139
	METEOR	CE	0.125	0.128	0.125	**0.132 (+ 2.08%)**	0.126	0.124	0.124	0.129	0.126	0.127	0.124
		KL	0.125	0.123	**0.129**	0.127	0.130	0.129	0.124	0.132	0.125	0.128	0.123

Table 1: Comparison of sacreBLEU, ROUGE L and METEOR scores using variants of Cross Entropy (CE) loss and Kullback–Leibler (KL) divergence loss on DailyDialog (DD) and EmpatheticDialogues (ED) datasets; s denotes amount of smoothing, where s ∈ (0.1, 0.2, NA); t = cosine similarity threshold, where t ∈ (0, 0.5, NA); w = apply synonym based filtering, where w ∈ (0, 1, NA).

with exact word matching. Table 1 summarizes our results, where the columns containing "NA" are the baseline results, against which improvements are measured. Further, Section A.1 (Appendix A) contains results for all configurations with additional evaluation metrics like BERTscore (Zhang* et al., 2020) and ROUGE 1 & 2.

Observations From the experiments we observe that, **(i)** Using a data dependent cosine similarity based distribution for label smoothing significantly outperforms the baseline (vanilla entropy based loss with or without label smoothing). We observe 12.67 % increase in BLEU score, 0.57 % increase in ROUGE L score, and 4.16 % increase in METEOR score for the DailyDialog dataset, and 1.42 % increase in BLEU score, 1.95 % increase in ROUGE L score, and 2.08 % increase in METEOR score for the EmpatheticDialogues dataset. **(ii)** Using additional WordNet synonym based filtering (w) does not help performance. To understand why this is happening, we plotted the distribution of the smoothing factor s for the randomly selected word "fun", and observed that the word had only one overlapping WordNet synonym in our vocabulary: "play". This caused the word "play" to be assigned a probability of 0.1, while all the other words are assigned a probability of 0, except for "fun", which was assigned a probability of 0.9. We reason that the sparsity in synonyms does not help in reducing the overconfidence of the model, as the final distribution is very similar to non-smoothing targets. Figure 2 illustrates the probabilities assigned to the incorrect labels of the word "fun", by each of the methods discussed in this paper. **(iii)** Using CE loss instead of KL generally improves performance while using label smoothing. We reason that this happens because in case of label smoothing, the constant entropy coefficient in KL loss reduces the overall loss, thus reducing the gradients during back propagation, which results in slower learning. **(iv)** Generally, using high smoothing value (s) does not help in learning. **(v)** The cosine similarity threshold t should be treated as a hyperparameter, and will require tuning depending on the vocabulary of the dataset used. **(vi)** We also noticed that a cosine similarity threshold t as high as 0.8 does not help in learning. We reason that using a high threshold creates a scenario similar to using WordNet synonyms, where the smoothing probability is distributed among very few (or no) words. Note that in order to enhance readability, the results with 0.8 threshold are omitted from Table 1, and are presented in the additional supplementary materials.

5 Conclusion

Label smoothing has an undesirable property of assigning uniform probability to incorrect labels, which present an incorrect knowledge to learn from. In this paper we propose ways to convert the uniform distribution to a data dependent distribution by weighing the smoothing probability using cosine similarity of word embeddings between the

correct and incorrect labels. We further experiment with WordNet synonyms as an additional filtering criteria, and report our findings. Although we achieve significant improvements over all baselines, we notice a drawback where the proposed system is unable factor in context while weighing the distribution of the incorrect labels. As future research, we intend to address this drawback using more contextualised representations instead of static embeddings.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. *arXiv preprint arXiv:2004.02594*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianxing He and James Glass. 2020. Negative training for neural dialogue response generation.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Shaojie Jiang, Pengjie Ren, Christof Monz, and Maarten de Rijke. 2019. Improving neural response diversity with frequency-aware cross-entropy loss. *The World Wide Web Conference on - WWW '19*.

Daniel Kang and Tatsunori Hashimoto. 2020. Improved natural language generation via loss truncation.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher 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, Doha, Qatar. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Guoyin Wang, Chunyuan Li, Jianqiao Li, Hao Fu, Yuh-Chen Lin, Liqun Chen, Yizhe Zhang, Chenyang Tao, Ruiyi Zhang, Wenlin Wang, Dinghan Shen, Qian Yang, and Lawrence Carin. 2020. Improving text generation with student-forcing optimal transport.

Yida Wang, Yinhe Zheng, Yong Jiang, and Minlie Huang. 2021. Diversifying dialog generation via adaptive label smoothing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3507–3520, Online. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Appendix

A.1 All Experiment Results

Table 2 shows the different variants of the baselines that were computed for both the DailyDialog and EmpatheticDialogues datasets. All performance improvements are compared against these baselines. For a metric, the best baseline score among all the hyperparameter settings is chosen to report improvements. Table 3 shows the results of using different hyperparameter settings and loss function in the DailyDialog dataset, and Table 4 shows the results obtained on the EmpatheticDialogues dataset. The best results with detailed comparison against baselines are already discussed in the main paper.

A.2 Model Training and Parameters

All the models were trained on a single Nvidia V-100 GPU, for 15 epochs each with a learning rate of 2e-4, batch size of 64, and using AdamW optimizer. The gradients of the model were clipped with a value of 1, and dropout with probability 0.1 was applied during training. The average runtime of each experiment is 60 minutes, with each of the trained models having 17.7 M parameters. The code, dataset and best performing models are publicly available through this link: download link.

		DailyDialog Dataset			EmpatheticDialogue Dataset		
		s = NA	**s = 0.1**	**s = 0.2**	**s = NA**	**s = 0.1**	**s = 0.2**
		t = NA	**t = NA**	**t = NA**	**t = NA**	**t = NA**	**t = NA**
Metric	**Loss**	**w = NA**	**w = NA**	**w = NA**	**w = NA**	**w = NA**	**w = NA**
sacreBLEU	CE	1.6625	1.8523	1.7251	2.2794	2.4084	2.1903
	KL	1.9469	1.7536	1.7931	2.2715	2.1682	2.2797
BERTScore	CE	0.8522	0.8520	0.8520	0.8539	0.8544	0.8527
	KL	0.8529	0.8520	0.8510	0.8540	0.8531	0.8541
ROUGE 1	CE	0.1272	0.1312	0.1319	0.1536	0.1592	0.1527
	KL	0.1336	0.1298	0.1300	0.1560	0.1545	0.1587
ROUGE 2	CE	0.0282	0.0303	0.0299	0.0251	0.0292	0.0251
	KL	0.0305	0.0283	0.0282	0.0267	0.0259	0.0271
ROUGE L	CE	0.1209	0.1243	0.1243	0.1382	0.1437	0.1373
	KL	0.1263	0.1223	0.1233	0.1406	0.1395	0.1426
METEOR	CE	0.1244	0.1324	0.1286	0.1254	0.1287	0.1257
	KL	0.1324	0.1303	0.1303	0.1250	0.1233	0.1297

Table 2: Baseline results of diverse automatic text generation metrics on the DailyDialog and EmpatheticDialogues datasets. The hyperparameters s, t and w control the usage of Label Smoothing, Cosine similarity threshold and WordNet filtering respectively. For the baseline, t and w were not used, which is indicated by NA. s = NA signifies vanilla entropy based loss without Label Smoothing.

		s = 0.1						**s = 0.2**					
		t = 0		**t = 0.5**		**t = 0.8**		**t = 0**		**t = 0.5**		**t = 0.8**	
Metric	**Loss**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**
sacreBLEU	CE	1.9896	1.7627	2.1936	1.8575	1.6676	1.8859	2.1158	1.8020	2.0536	1.9302	1.5674	1.8502
	KL	1.8459	1.8181	1.9128	1.8858	1.7957	1.7453	1.9387	1.8092	1.7292	1.8856	1.5874	1.9707
BERTScore	CE	0.8518	0.8529	0.8515	0.8527	0.8507	0.8509	0.8513	0.8525	0.8525	0.8519	0.8507	0.8512
	KL	0.8520	0.8527	0.8517	0.8515	0.8520	0.8516	0.8509	0.8525	0.8522	0.8518	0.8518	0.8515
ROUGE 1	CE	0.1309	0.1279	0.1353	0.1326	0.1260	0.1280	0.1298	0.1271	0.1315	0.1317	0.1250	0.1290
	KL	0.1301	0.1332	0.1318	0.1311	0.1281	0.1276	0.1301	0.1328	0.1310	0.1312	0.1263	0.1325
ROUGE 2	CE	0.0282	0.0276	0.0309	0.0300	0.0287	0.0280	0.0286	0.0286	0.0308	0.0310	0.0276	0.0305
	KL	0.0283	0.0312	0.0300	0.0294	0.0297	0.0291	0.0285	0.0312	0.0292	0.0299	0.0277	0.0299
ROUGE L	CE	0.1238	0.1209	0.1270	0.1260	0.1200	0.1203	0.1217	0.1204	0.1238	0.1244	0.1183	0.1222
	KL	0.1227	0.1264	0.1243	0.1242	0.1213	0.1207	0.1223	0.1253	0.1232	0.1234	0.1185	0.1252
METEOR	CE	0.1342	0.1287	0.1379	0.1314	0.1270	0.1319	0.1344	0.1279	0.1346	0.1313	0.1223	0.1280
	KL	0.1346	0.1324	0.1327	0.1311	0.1262	0.1275	0.1319	0.1310	0.1298	0.1296	0.1247	0.1330

Table 3: Results of diverse automatic text generation metrics on the DailyDialog dataset, trained with variants of Entropy based loss with different hyperparameter settings: cosine similarity threshold (t), Label Smoothing (s) and WordNet filtering (w).

		s = 0.1						**s = 0.2**					
		t = 0		**t = 0.5**		**t = 0.8**		**t = 0**		**t = 0.5**		**t = 0.8**	
Metric	**Loss**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**	**w = 0**	**w = 1**
sacreBLEU	CE	2.4427	2.2082	2.1922	2.2164	2.3467	2.2596	2.3187	2.2622	2.3125	2.2569	2.3944	2.2767
	KL	2.2774	2.2781	2.3370	2.3615	2.2347	2.2769	2.4319	2.2749	2.4393	2.1431	2.2566	2.2652
BERTScore	CE	0.8543	0.8539	0.8547	0.8528	0.8536	0.8544	0.8544	0.8536	0.8539	0.8532	0.8531	0.8544
	KL	0.8541	0.8543	0.8544	0.8528	0.8536	0.8528	0.8544	0.8528	0.8544	0.8526	0.8535	0.8543
ROUGE 1	CE	0.1612	0.1564	0.1577	0.1531	0.1558	0.1551	0.1589	0.1550	0.1575	0.1531	0.1553	0.1590
	KL	0.1594	0.1613	0.1596	0.1540	0.1549	0.1552	0.1619	0.1554	0.1588	0.1545	0.1564	0.1569
ROUGE 2	CE	0.0287	0.0270	0.0271	0.0250	0.0274	0.0267	0.0269	0.0265	0.0267	0.0264	0.0261	0.0262
	KL	0.0270	0.0290	0.0273	0.0266	0.0256	0.0253	0.0288	0.0269	0.0274	0.0257	0.0251	0.0268
ROUGE L	CE	0.1443	0.1409	0.1425	0.1385	0.1402	0.1388	0.1416	0.1398	0.1411	0.1381	0.1396	0.1423
	KL	0.1441	0.1454	0.1435	0.1387	0.1397	0.1393	0.1465	0.1401	0.1430	0.1394	0.1404	0.1416
METEOR	CE	0.1324	0.1266	0.1248	0.1245	0.1267	0.1264	0.1291	0.1266	0.1278	0.1243	0.1247	0.1292
	KL	0.1272	0.1302	0.1290	0.1246	0.1235	0.1254	0.1323	0.1253	0.1283	0.1234	0.1257	0.1269

Table 4: Results of diverse automatic text generation metrics on the EmpatheticDialogues dataset, trained with variants of Entropy based loss with different hyperparameter settings: cosine similarity threshold (t), Label Smoothing (s) and WordNet filtering (w).