Cformer: Semi-Supervised Text Clustering Based on Pseudo Labeling

Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, Frank Drewes
Department of Computing Science, Umeå University, Sweden
{arezooh,sonvx,monowar,drewes}@cs.umu.se

ABSTRACT

We propose a semi-supervised learning method called *Cformer* for automatic clustering of text documents in cases where clusters are described by a small number of labeled examples, while the majority of training examples are unlabeled. We motivate this setting with an application in contextual programmatic advertising, a type of content placement on news pages that does not exploit personal information about visitors but relies on the availability of a high-quality clustering computed on the basis of a small number of labeled samples.

To enable text clustering with little training data, Cformer leverages the teacher-student architecture of Meta Pseudo Labels. In addition to unlabeled data, Cformer uses a small amount of labeled data to describe the clusters aimed at. Our experimental results confirm that the performance of the proposed model improves the state-of-the-art if a reasonable amount of labeled data is available. The models are comparatively small and suitable for deployment in constrained environments with limited computing resources. The source code is available at https://github.com/Aha6988/Cformer.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; **Neural networks**; **Semi-supervised learning settings**; • **Information systems** → **Computational advertising**; **Clustering**.

KEYWORDS

meta pseudo clustering, semi-supervised learning, pseudo labeling

ACM Reference Format:
Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, Frank Drewes. 2021. Cformer: Semi-Supervised Text Clustering Based on Pseudo Labeling. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482073

1 INTRODUCTION

Clustering in its purest form refers to unsupervised methods for dividing a set of n data points into k so-called clusters, groups of closely related points. For this, a similarity measure between data points is required. When the objective is to cluster text documents, using the similarity of document vectors given by some standard model usually does not work very well because of the high dimensionality of these vector spaces [1]. Furthermore, downstream tasks often require the clusters to carry meaning. An application area in which this is the case is the one that motivated this work: contextual programmatic advertising. To make clusters reflect intended meaning, one would ideally want the clustering approach to be trained on labeled data. Unfortunately, this area is also one in which large amounts of labeled data are hard to come by.

In programmatic advertising [12], the aim is to fill ad space on, e.g., a news page, in real time with suitable ads when a visitor of the site accesses the page. To accomplish this, programmatic advertising platforms conduct auctions for ad space the moment pages are accessed. Software agents representing advertisers place their bids according to their notion of how much the advertising space is worth, and the ad space goes to the one who wins the auction. The worth of the ad space is traditionally estimated based on personal information about the visitor, such as their viewing history. Contextual advertising is a comparatively new idea that

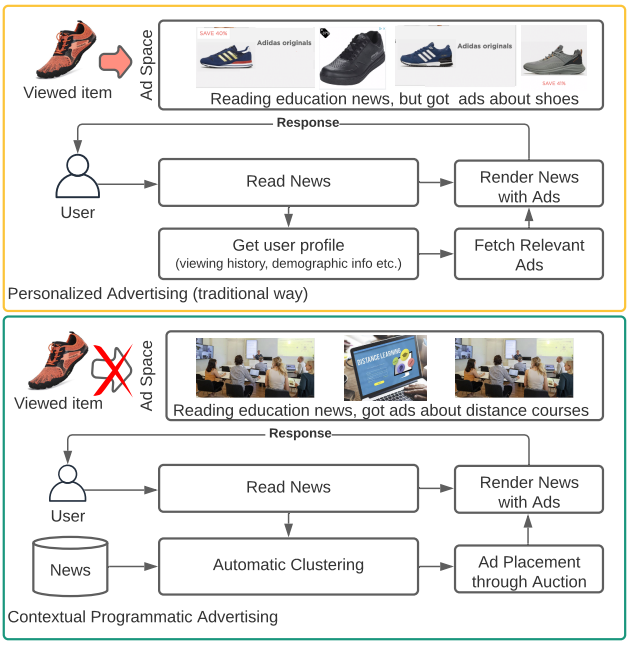

Figure 1: A conceptual comparison of personalized and contextual advertising. The former exploits personal information, the latter uses only the news content, thus being less intrusive.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482073

challenges this model. It avoids the use of personal information for both privacy and efficiency reasons by focusing on the content of the news page to decide how well the ad fits it.[1] Here, "fitting" often does not simply mean that the contents of news article and ad align. Companies often conduct advertising campaigns during which they want their ads to be seen (or not to be seen) in contexts that promote a certain image, regardless of the specific product being advertized.

Abstractly, each desired context can be understood as a cluster. These clusters and their descriptions change over time as campaigns are canceled and new ones are set up. Most importantly, as campaigns may focus on arbitrary aspects, there is typically little labeled data available. To cope with this situation, we propose *Cformer*, a semi-supervised clustering approach that makes use of a small amount of labeled documents (news articles provided as typical example contexts for a given advertising campaign) and a larger number of unlabeled documents (uncategorized news articles).

Cformer is inspired by the recent work of Pham et al. [11] on meta pseudo labels, an extension of pseudo labeling. The latter is a successful semi-supervised learning method which resulted in state-of-the-art performance in many computer vision tasks. It works by having a pair of networks, a *student* and a *teacher*. The teacher model predicts labels for unlabeled data, so-called pseudo labels. Then both pseudo labeled data and the original labeled data are used to train the student. To tackle the confirmation bias (the student learns to confirm the teacher), the idea of meta pseudo labeling is to train teacher and student in parallel, letting the teacher use the performance of the student on labeled data to predict better pseudo labels. We transfer this idea to the realm of text clustering. Also, our Distill-Cformer model departs from using identical teacher and student architectures. This speeds up training, which is important for contextual advertising due to the frequently changing campaigns.

The main contributions of the present work are:

- The proposed Cformer model utilizes meta pseudo labels for document clustering. The architecture is adaptable to similar tasks such as document classification and document retrieval.
- We further introduce Distill-Cformer to confirm the effectiveness of our proposed architecture on a much smaller neural model (i.e., DistilBert [13]) for the student, thus considerably reducing the overall training time.
- We conduct performance tests with two benchmark datasets. The results of these experiments indicate that Cformer and Distill-Cformer outperform the state of the art in most cases.

2 RELATED WORK

Previous work on contextual advertising tried to exploit prior knowledge (usually in the form of labeled words for each class) or generating labeled data automatically. Jin et al. [6] model contextual targeting as a lightly-supervised one-class classification problem. Their algorithm takes unlabeled documents and the labeled keywords for the target class c as input and returns a classifier M_c identifying documents that belong to class c. Jin et al. [5] automatically map the categories in the Interactive Advertising Bureau (IAB) taxonomy to category nodes in the Wikipedia category graph and propagate labels across the graph to obtain a list of labeled Wikipedia documents for training purposes.

We tackle document clustering with limited labeled data by semi-supervised learning. Such methods add more flexibility to supervised approaches by needing only a very small portion of the dataset to be labeled. Many of the recent approaches in semi-supervised learning use consistency training on a large amount of unlabeled data [7, 14]. These methods regularize model predictions to be invariant to small levels of noise. Data augmentation methods are used to enlarge labeled datasets in supervised learning cases when training data is not sufficient, e.g., in Augmented SBERT [15]. Further, these methods can be used to inject noise to data. Xie et al. [16] investigate the role of noise injection in consistency training and propose Unsupervised Data Augmentation (UDA) to replace the traditional noise injection methods by high quality data augmentation such as back translation of textual data. Chen et al. [3] propose the data augmentation method TMix that takes in two text instances and interpolates them in their corresponding hidden space. Based on TMix they propose MixText, a semi-supervised learning method for text classification and clustering. MixText predicts labels for unlabeled data and then uses TMix to interpolate between labeled and unlabeled data to impose a regularization on the model.

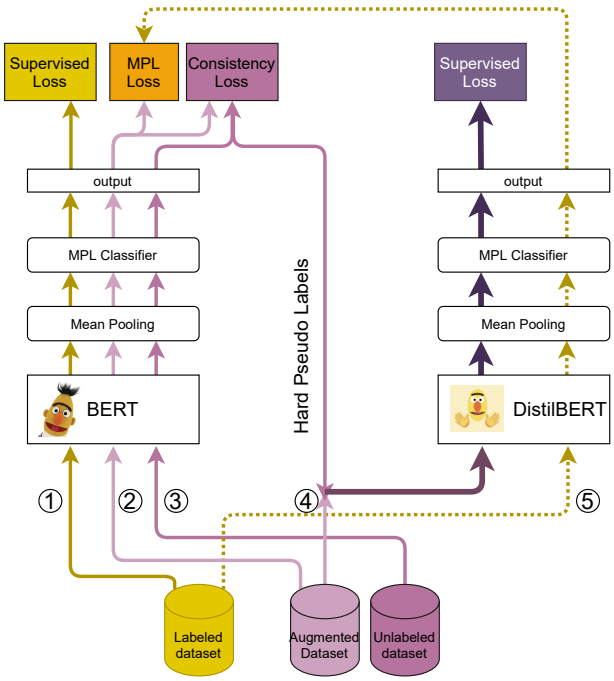

Figure 2: The teacher(left network)-student(right network) architecture of Distill-Cformer.

3 METHODOLOGY

Figure 2 shows our proposed approach for semi-supervised clustering. Following the architecture of [11], we have a *teacher* model

[1] If someone has recently bought new shoes but they are currently looking at a news page about self-education, they might not be interested in buying yet another pair of shoes, but would perhaps be inclined to sign up for online courses (see Figure 1).

T with learnable parameters Θ_T (left side in Figure 2) and a *student* model S with learnable parameters Θ_S (right side in Figure 2) that are trained in parallel. The teacher is trained with the Unsupervised Data Augmentation (UDA) objective [16] and feedback from the student [11]. The UDA objective consists of supervised loss on labeled data and consistency loss between unlabeled and augmented data. Additional feedback is the performance of the student on labeled data (which is assumed to be correctly labeled). The student is trained with supervised loss on the pseudo labeled data provided by the teacher. Augmented data is built by applying text augmentation techniques (e.g., word substitution with the most suitable word found by ContextualWordEmbsAug [9]) on unlabeled data. As Figure 2 shows, both the student and the teacher consist of encoders that map documents to their distributed representations (transformer and a mean pooling module that computes the average of the transformer outputs in different positions) followed by a classifier.

In a first training step, a batch of labeled data (x_l, y_l) (track ① in Figure 2), a batch of unlabeled data x_u (track ③), and its augmented version x_a (track ②) are fed to the teacher. The cross entropy loss is computed between labels y_l and teacher outputs for x_l:

$$Loss_T^l = CrossEntropy(y_l, T(x_l; \theta_T)) \ .$$

Unsupervised or consistency loss is computed using x_u and x_a. The consistency loss constrains the model predictions to be invariant to input noise by forcing augmented samples to have the same labels as the original data samples. Moreover, to encourage the model to predict confident low-entropy labels for unlabeled data, we use a sharpening function over soft predictions for x_u denoted as y_u^{soft}. We utilize the sharpening function used in Chen et al. [3]. Given *soft pseudo labels* y_u^{soft} and a temperature hyper-parameter *Temp*

$$y_u^{soft} = T(x_u; \Theta_T)$$
$$y_u^{sharp} = sharpen(y_u^{soft}, Temp) = \frac{(y_u^{soft})^{\frac{1}{T}}}{\|(y_u^{soft})^{\frac{1}{T}}\|}$$

where $\|.\|$ is the l_1-norm of the vector. So, the teacher unsupervised loss is

$$Loss_T^u = CrossEntropy(y_u^{sharp}, T(x_a; \Theta_T)) \ .$$

We found it helpful to mask out examples that the current model is not confident about. So, in each batch, the consistency loss term is computed only on examples whose highest probability among clustering categories is greater than an experimentally determined threshold β.

In a second step, the student model learns from pseudo labeled data annotated by the teacher. The augmented batch x_a (as a regularization to make the student insensitive to noise) and *hard pseudo labels* y_u^{hard} (cross-point ④ in Figure 2) are fed to the student. The student tries to minimize the cross entropy loss between the hard pseudo labels and its own predictions. The hard pseudo labels y_u^{hard} are generated by considering the clusters with the highest values among the soft pseudo labels y_u^{soft} as the correct clusters. Therefore:

$$y_u^{hard} = j : y_{u,j}^{soft} = \max_i(y_{u,i}^{soft})$$
$$Loss_S^l = CrossEntropy(y_u^{hard}, S(x_a; \Theta_S)) \ .$$

In parallel, the teacher learns from the reward signal of how well the student performs on labeled data (x_l, y_l) (dotted line ⑤ from student to teacher in Figure 2). This loss is called *Meta Pseudo Labels (MPL) loss*. Using the parameters of the student after updating with $Loss_S^l$ as $\Theta_S^{'}$:

$$Loss_T^{MPL} = \nabla_{\Theta_T} CrossEntropy(y_l, S(x_l; \Theta_S^{'})) \ .$$

To see how this loss is exactly computed and its derivation equations, we refer to Pham et al. [11].

Combining the three losses, we get the overall objective function of the teacher:

$$Loss_T = Loss_T^l + \lambda_u * loss_T^u + Loss_T^{MPL}$$

where λ_u is the contribution coefficient of the consistency loss.

Finally, as the student only learns from unlabeled data with pseudo labels generated by the teacher, we fine-tune the student (that has converged after training with pseudo labels) on labeled data to improve its accuracy. Moreover, to increase the generalization capability of both student and teacher, we use label smoothing [10] when computing supervised losses $Loss_T^l$ and $Loss_S^l$ to prevent the model from overfitting to labeled data.

4 EXPERIMENTS AND RESULT ANALYSIS

We perform experiments with two English text classification benchmark datasets: AG News [17] and Yahoo! answers [2]. For Yahoo! answers, we obtain the text to be clustered by concatenating the question title, question content and best answer; for AG News we only utilize the news content (without titles). To be comparable with our baselines, we randomly sample the same amount of data as in [3] from the original training sets for our unlabeled and validation sets and used the original test sets. The dataset statistics and splits are available in Table 1. To generate augmented data from unlabeled data, we use the library *nlpaug* [9]. We substitute text words based on contextual word embeddings with probability 0.9.

We use MixText [3] together with two of its baselines (BERT [4] and UDA [16]) as our baseline models and compare our results against the results for these models as reported in [3]. The BERT baseline is a BERT-base-uncased model fine-tuned only with the labeled data for text classification. It consists of a two-layer MLP (as in our model) on top of the BERT encoder. The UDA baseline is a PyTorch version of the original UDA model implemented for GPU by the inventors of MixText.

We consider two variations of our proposed architecture with different encoder components:

(1) **Cformer** model: the student and the teacher models both use the BERT-base-uncased model.
(2) **Distill-Cformer** model: the teacher is the same as in Cformer but the student uses DistilBERT-base-uncased.

The teacher and student models in Cformer have 109.58 million parameters; the student in Distill-Cformer has 66.46 million parameters. We use the BERT-based-uncased tokenizer to tokenize the text, average pooling over the output of the transformer to aggregate word embeddings into document embedding, and a two-layer MLP with a 128 hidden size and hyperbolic tangent as its activation function (the same as in MixText) to predict the labels. Documents are truncated to their first 256 tokens. Like UDA and MixText, in all

Table 1: Dataset statistics and dataset split. The number of sentences and words are denoted by #s and #w, respectively. The number of unlabeled, dev, and test data items are given in terms of the number of data items per class.

Dataset	Classes	Documents	Average #s	Max #s	Average #w	Max #w	Vocabulary	Unlabeled	Dev	Test
Yahoo! answers	10	1,450,000	6.4	515	108.4	4002	1,554,607	5000	5000	6000
AG News	4	120,000	1.7	20	36.2	212	94,443	5000	2000	1900

Table 2: Experimental results of our proposal models (Cformer & Distill-Cformer) in comparison with SoTA models. Bold values indicate the highest performance per column.

Dataset	Model	10	200	2500	Dataset	Model	10	200	2500
	BERT	69.5	87.5	90.8		BERT	56.2	69.3	73.2
	UDA [16]	84.4	88.3	91.2		UDA [16]	63.2	70.2	73.6
AG News	MixText [3]	88.4	89.2	91.5	**Yahoo! answers**	MixText [3]	**67.6**	71.3	74.1
	Cformer (Ours)	**88.7**	89.9	91.8		Cformer (Ours)	66.8	**72.0**	**74.5**
	Distill-Cformer (Ours)	88.0	**90.0**	**91.9**		Distill-Cformer (Ours)	65.2	71.9	74.3

experiments, the labeled and unlabeled batch sizes are 4 and 8, respectively. Both models are trained with the AdamW optimizer [8]. We train our models for 7000 steps (including 50 warm-up steps) and evaluate them every 500 steps. To avoid overfitting, we use early stopping with delta 5E-3 and patience 4. We set the learning rate of the transformer and classifier components in both models to 1E-5 and 1E-3 respectively. After training both models, we fine-tune the student on the labeled dataset using the AdamW optimizer with a fixed learning rate of 5E-6 and a batch size of 32, running for 10 epochs. The temperature T for sharpening is set to 0.5 for Yahoo answers and 0.3 for AG News. The confidence threshold β is set to 0.9 and the label smoothing parameter is 0.15 for both datasets. For the contribution coefficient of unsupervised loss in the teacher loss function λ_u, we start from 0 and increase it linearly for 6000 steps until it reaches 1. All experiments are run using 4 GPU V100 32GB. With small batch sizes, the model can be trained using other regular GPUs. Since we kept the same batch size as previous work, the training process only occupies 16GB of memory per GPU.

4.1 Result Analysis

Table 2 presents our results with Cformer and Distill-Cformer in comparison with other methods.

Overall performance of Cformer. In comparison with the current state-of-the-art models, we can observe that ours yield good performance across the considered datasets. First, our model outperformed UDA in all experiments. In fact, Cformer achieves better accuracy than UDA from 0.6% to more than 4% across these datasets. Since the teacher in our model is trained with the UDA objective function, this shows the effectiveness of using pseudo labels and knowledge distillation from teacher to student. Second, in comparison to MixText, Cformer stably works better on both datasets unless the number of labeled samples is very small. For 10-shot cases, Cformer achieves better performance on AG News but worse on Yahoo! answers. In this regard, it is worth observing that the AG News dataset is easier to learn than the Yahoo! answers dataset, due to its smaller vocabulary and the smaller number of documents. Therefore, less labeled data is required to learn how to classify AG news than for the Yahoo! answers dataset.

Cformer vs. Distill-Cformer. Given the requirement of getting high performance in constrained environments, we are especially interested in analyzing Distill-Cformer. Generally, it exhibits on-par performance compared to Cformer even though its student model is considerably smaller. The gap in performance is less than 0.2% in most cases. This indicates that the size of the model is not a bottleneck as long as the knowledge distillation works effectively. Moreover, Distil-Cformer offers faster inference time than Cformer since its architecture is smaller in size. Specifically, testing on the Test set of *Yahoo! answers* dataset with one GPU, the inference time of Distil-Cformer was *143.5 seconds*, which is 2 times faster than that of Cformer (*287.0 seconds*). This result again confirmed the finding of Sanh et al. [13] pointing out that "DistilBERT retains 97% of the performance with 40% fewer parameters".

5 CONCLUSION

We have presented Cformer, a teacher-student architecture for semi-supervised text clustering in contexts where clusters are given by a limited number of labeled samples. An example application is dynamic content placement on contextual advertising platforms. In general, we expect the technique to be useful for all downstream tasks which require text classification based on partially labeled training data, especially when the labels and the amount of labeled data change over time (as in the case of advertising campaigns).

Cformer showcases a new approach in dealing with both short and long texts datasets. It effectively performs better on both short text data (AG News) and long text data (Yahoo! answers) by integrating the knowledge distillation into the learning process. Moreover, the proposed models work effectively on both full-size BERT and DistillBERT as the encoders. Cformer outperforms the state-of-the-art approaches with various settings, especially when sufficient labeled data is available. For applications such as content placement on web pages, a useful extension would be a multimodal version (e.g., Zong et al. [18] on multimodal clustering).

REFERENCES

[1] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*. Springer, 420–434.

[2] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of Semantic Representation: Dataless Classification.. In *Aaai*, Vol. 2. 830–835.

[3] Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2147–2157. https://doi.org/10.18653/v1/2020.acl-main.194

[4] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[5] Yiping Jin, Vishakha Kadam, and Dittaya Wanvarie. 2021. Bootstrapping Large-Scale Fine-Grained Contextual Advertising Classifier from Wikipedia. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*. Association for Computational Linguistics, Mexico City, Mexico, 1–9. https://aclanthology.org/2021.textgraphs-1.1

[6] Yiping Jin, Dittaya Wanvarie, and Phu Le. 2017. Combining lightly-supervised text classification models for accurate contextual advertising. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 545–554.

[7] Samuli Laine and Timo Aila. 2017. Temporal Ensembling for Semi-Supervised Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. https://openreview.net/forum?id=BJ6oOfqge

[8] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. https://openreview.net/forum?id=Bkg6RiCqY7

[9] Edward Ma. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug.

[10] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help?. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 4696–4705.

[11] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. 2021. Meta Pseudo Labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11557–11568.

[12] Anthony Samuel, Gareth RT White, Robert Thomas, and Paul Jones. 2021. Programmatic advertising: An exegesis of consumer concerns. *Computers in Human Behavior* 116 (2021), 106657.

[13] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. (2019). http://arxiv.org/abs/1910.01108

[14] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 1195–1204.

[15] Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 296–310. https://doi.org/10.18653/v1/2021.naacl-main.28

[16] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised Data Augmentation for Consistency Training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, December 6-12, 2020, virtual*.

[17] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 649–657.

[18] Linlin Zong, Faqiang Miao, Xianchao Zhang, and Bo Xu. 2020. Multimodal Clustering via Deep Commonness and Uniqueness Mining. In *The 29th ACM International Conference on Information and Knowledge Management (CIKM), Virtual Event, October 19-23, Ireland*. ACM, 2357–2360. https://doi.org/10.1145/3340531.3412103