CC-Top: Constrained Clustering for Dynamic Topic Discovery

Jann Goschenhofer[1,2♠] **Pranav Ragupathy**[1♣] **Christian Heumann**[1♠] **Bernd Bischl**[1,2,3♠]
Matthias Aßenmacher[1♠]
[1] Department of Statistics, LMU, Munich, Germany
[2] Fraunhofer IIS, Erlangen, Germany
[3] Munich Center for Machine Learning (MCML), LMU, Munich, Germany
♠{jann.goschenhofer,chris,bernd.bischl,matthias}@stat.uni-muenchen.de
♣p.ragupathy@campus.lmu.de

Abstract

Research on multi-class text classification of short texts mainly focuses on supervised (transfer) learning approaches, requiring a finite set of pre-defined classes which is constant over time. This work explores deep constrained clustering (CC) as an alternative to supervised learning approaches in a setting with a dynamically changing number of classes, a task we introduce as *dynamic topic discovery* (DTD). We do so by using pairwise similarity constraints instead of instance-level class labels which allow for a flexible number of classes while exhibiting a competitive performance compared to supervised approaches. First, we substantiate this through a series of experiments and show that CC algorithms exhibit a predictive performance similar to state-of-the-art supervised learning algorithms while requiring less annotation effort. Second, we demonstrate the overclustering capabilities of deep CC for detecting topics in short text data sets in the absence of the ground truth class cardinality during model training. Third, we showcase how these capabilities can be leveraged for the DTD setting as a step towards dynamic learning over time. Finally, we release our codebase to nurture further research in this area.

1 Introduction

There has been substantial research on methods for the classification of short user-generated texts such as customer reviews, search queries, tweets, or articles (Mohammad et al., 2016; Sun et al., 2019; Barbieri et al., 2020). Often, despite being handled differently in supervised frameworks, one does not know *a-priori* what these classes are, how many there are at time point t, or how many there will be at a future time point $t+1$. In existing benchmark data sets from the natural language processing (NLP) research community (e.g. Lang, 1995; Lehmann et al., 2015), this potential issue is largely ignored, since only one training set is provided alongside one test set. Performance can thus only be measured in a static fashion, i.e. for one fixed time point. While this problem of an unknown number of classes is often tackled using unsupervised learning techniques (Deerwester et al., 1990; Blei et al., 2003), these algorithms come with an array of limitations and are not able to (automatically) adapt to a changing number of classes. We formally introduce this novel problem setting with dynamically changing topics as DTD and explore the potential of deep constrained clustering (CC; Hsu et al., 2019) algorithms coupled with pre-trained language models (BERT; Devlin et al., 2019) for text classification in this setting.

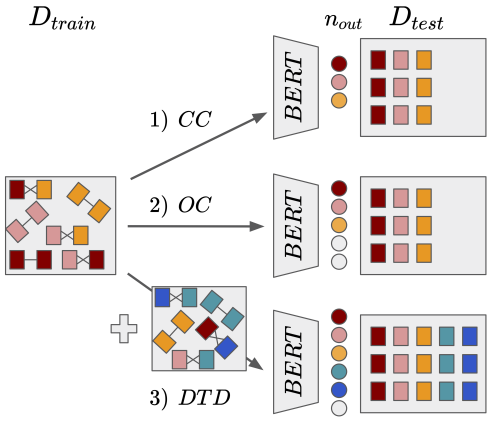

Figure 1: Illustration of CC-Top and the training paradigms 1) constrained clustering (CC), 2) overclustering (OC) and 3) dynamic topic discovery (DTD). Crosses and lines represent Cannot- and Must-Link pairwise relations, respectively.

Various approaches have been developed to combine CC (Wagstaff and Cardie, 2000) with neural networks, mainly for image datasets (Hsu and Kira, 2015; Hsu et al., 2019). In addition to strong predictive clustering performance, these methods are able to recover the number of distinct clusters in the data without access to instance-level class labels during training. Hence, they can be used for category detection, a capability that we leverage for the detection of dynamically changing topics.

Proceedings of the The First Workshop on Ever Evolving NLP (EvoNLP), pages 26 - 34

Moreover, they address and alleviate the problem of label annotation: Human annotators only need to annotate pairs of samples indicating whether they belong to a similar topic or not instead of annotating one distinct class label per sample. We argue that for short texts this is easier and more efficient than annotating individual samples.

We propose the use of **C**onstrained **C**lustering for **Top**ic classification (CC-Top, cf. Fig. 1): We 1) leverage pairwise constraint annotations for topic classification of short texts in a weakly supervised manner, we 2) demonstrate its topic discovery capabilities and 3) introduce a new problem setting with dynamically changing topics. In a series of experiments, we substantiate these findings and publish our codebase[1] to nurture further research on constrained clustering in the NLP community.

2 Related Work

With the advent of supervised fine-tuning of pre-trained models, text clustering performance further increased (Huang et al., 2020; Schopf et al., 2021). One main limitation of these models is their dependence on a given amount of clusters as input for model training, which limits their use for the detection of clusters, i.e., topics/classes. Unsupervised topic modeling algorithms (e.g. Blei et al., 2003; Grootendorst, 2022) are no real alternative here, since we focus on topic *classification* and not on topic *modeling*. Note, that we make a clear distinction between these two approaches here: Topic modeling aims at uncovering latent structures in the data and puts a large emphasis on explaining and interpreting the detected clusters. Further, as opposed to *Topic classification*, it does not assume the cluster assignment to be mutually exclusive, i.e. a document is regarded as a (potential) mixture of multiple topics. Since this is in sharp contrast to the setting we are investigating, we do not consider such approaches as potential unsupervised baselines.

In turn, CC allows this detection of the number of clusters using binary pairwise constraint annotations. The introduction of pairwise constraints for clustering (Wagstaff and Cardie, 2000) led to the adaptation of existing clustering methods towards the use of constraints (Basu et al., 2004) (see Gançarski et al. (2020) for an overview). With the proposal of the KCL loss based on the Kullback-Leibler divergence, Hsu and Kira (2016) introduced CC to deep learning settings. They further showed its applicability to transfer learning (Hsu et al., 2018), introduced the MCL as an alternative loss (Hsu et al., 2019), and showed its applicability for cluster detection, i.e., overclustering. We use these two pairwise loss functions.

[1] https://github.com/rpranav22/cc-top

3 Materials and Methods

3.1 Method

We consider a dataset $\mathcal{D}$ that contains n_c constraint pairs of the form $x_{ij} = (x_i, x_j, c_{ij}) \in \mathcal{D}^c$, where x_i, x_j are two input samples and $c_{ij} \in \{0, 1\}$ is the associated binary constraint describing whether the samples are in the same ($c_{ij} = 1$, *Must-Link*) or different clusters ($c_{ij} = 0$, *Cannot-Link*). We refer to true class labels as $y_i \in \mathcal{Y}$, where $K = |\mathcal{Y}|$ describes the number of true underlying classes K in the data set. When K is not known, the model's number of output neurons n_{out} may differ from K. We train a deep CC model f with its final head consisting of a softmax layer i.e., the model predicts a probability distribution over cluster assignments $\hat{y}_i = f(x_i)$, where $\hat{y}_{il}$ denotes the predicted probability of x_i belonging to cluster $l \in 1, ..., n_{out}$.

We follow Hsu and Kira (2016); Hsu et al. (2019) for the training of the CC model: the model predictions $\hat{y}_i, \hat{y}_j$ for text samples x_i, x_j are fed into a pairwise loss function with their associated constraint c_{ij}. There exists a variety of loss functions that can deal with pairwise constraints (Zhang et al., 2021b), with the KCL (Hsu and Kira, 2016) and the MCL (Hsu et al., 2019) being the most prominent ones. The KCL is a pairwise loss function based on the Kullback-Leibler divergence between the pairwise model assignments $\hat{y}_i, \hat{y}_j$. Similarly, the MCL loss is aligned on the binary cross entropy loss and reportedly enables smoother model training. Following prior work (Lin et al., 2020; Zhang et al., 2021a), we use BERT (Devlin et al., 2019) as a language model backbone for f.[2] Note that throughout our experiments we randomly subsample a training dataset of $20,000$ pairwise constraints from the original fully labeled dataset.

Next to the application in settings where the true number of clusters K is known a-priori, CC models can also be used when this information is absent during model training. This is also referred to as overclustering (OC) where the model can

[2] Note that any (pre-trained) architecture can be used as a backbone in conjunction with these loss functions. All configurations can be found in Table 5 in Appendix A.

assign more clusters than present in the data, i.e. $n_{out} > K$. This capability to learn the number of clusters in the data from constraint annotations differentiates CC from clustering methods such as k-means, where K needs to be provided as a hyperparameter to the model, or supervised approaches.

3.2 Baselines

As a lower, unsupervised baseline, we use BERT embeddings combined with K-MEANS++ (Arthur and Vassilvitskii, 2006). For the fully supervised upper bound trained via instance-level class labels, we finetune the BERT-BASE-UNCASED architecture from huggingface (Wolf et al., 2020), following the standard pretrain-finetune paradigm. Both baselines are trained on the entire training dataset.

3.3 Dynamic Topic Discovery (DTD)

We now consider the scenario, where the set of classes is not fixed and known *a-priori* at time point t but is dynamically changing over time ($t+1, t+2, \dots$): First, at t, we have pairwise annotations for samples that belong to K_t distinct classes. Second, we train a CC model f_t to assign any new data point to one of the discovered clusters. Third, at $t+1$, we obtain new samples that could either belong to one of the initial K_t classes or to new, unseen classes and the model fails to classify the new samples accurately.

If our model was fully supervised (i.e., trained on instance-level class labels), we would have to reconsider the entire labeling scheme (i.e., produce the new classes and revisit all existing labeled samples from t) and re-train the entire model. However, in the case of CC, we can continue annotating the data using pairwise constraints and continue to train the existing model (i.e., let the model determine (i) if there are new classes and (ii) how many of them). We construct the following scenario to investigate the model's capability to adapt to a changing number of classes over time: First, we fix the architectural setup to CC-KCL on DBpedia and use $n_{out} = 30$ to provide the model with enough overclustering flexibility. Second, for $t = 1$, we take a subset of the training set, consisting of samples from 10 classes only, and sample $n_c = 20,000$ constraints from this subset, resulting in $38,056$ samples from 10 classes for training ($D_{train,t=1}$). Third, for $t = 2$, we select $18,000$ samples from the remainder of the training set ($D_{train,t=2}$) controlling for the ratio of the classes that the samples belong to $x\%$ from the 'old' 10 classes at $t = 1$ and $(100-x)\%$ from the 'new' 4 classes at $t = 2$, which were withheld from $D_{train,t=1}$. The DBpedia test set is also split into two distinct parts: $D_{test,1}$ contains only samples from the 10 'old' classes, and $D_{test,2}$ contains only samples from the 4 'new' ones. During the DTD experiments, we denote the entire test set as $D_{test,combined}$.

3.4 Datasets

We run experiments on three English datasets of short texts with associated instance-level class labels. An overview of the analyzed data sets AG News (Zhang et al., 2015), TREC coarse (Li and Roth, 2002), and DBpedia (Lehmann et al., 2015) is provided in Table 1. We did not perform any further special preprocessing. We used only DBPedia for further experiments with respect to DTD, since the number of classes in the other two data sets was too small to construct a meaningful DTD scenario.

Name	K	$\#Train$	$\#Val$	$\#Test$	Avg. Length
AG News	4	120,000	8,000	7,600	40
TREC coarse	6	4,952	500	500	10
DBpedia	14	560,000	35,000	35,000	50

Table 1: Overview of the data sets used for evaluation.

3.5 Performance Metrics

Following prior work (Hsu et al., 2019; Lin et al., 2020), we report model performance as measured in Accuracy (ACC), Normalized Mutual Information (NMI; Strehl and Ghosh, 2002) and the Adjusted Rand Index (ARI; Steinley, 2004). For more in-depth explanations and for the formulas of all three metrics, please refer to Appendix C. All three metrics are normalized to $[0, 1]$, where higher values indicate better performance. Similarly, we use the Hungarian algorithm (Kuhn, 1955) to optimally map predicted labels to the true cluster assignments before calculating the performance metrics.

4 Experiments

In Table 2, we compare the CC models trained via both the MCL and the KCL loss with the lower and upper baselines. These results confirm that CC is a suitable method to train weakly supervised models for the detection of topics in short texts, reaching almost full supervision performance.

Furthermore, we investigated the capabilities of these models in the OC scenario, where the ground truth number of classes is unknown during training and the model can potentially assign $n_{out} = 30 >$

		Lower Baseline			CC-KCL			CC-MCL			Upper Baseline		
Data set	K	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
AG News	4	0.830	0.577	0.605	0.870	0.714	0.739	**0.917**	**0.755**	**0.795**	0.919	0.759	0.800
TREC-coarse	6	0.542	0.299	0.302	0.953	0.890	0.900	**0.967**	**0.908**	**0.923**	0.962	0.897	0.917
DBPedia	14	0.631	0.726	0.494	**0.982**	**0.963**	**0.967**	0.661	0.805	0.653	0.989	0.974	0.977

Table 2: Averaged results for the baselines on all available training samples as well as for CC-MCL and CC-KCL trained with $20,000$ constraints each. The better CC model (between KCL and MCL) is marked in bold and CC models almost reach full supervision level performance (upper baseline). Refer to a larger version of this table including standard deviations across runs in Appendix B, Table 6.

Dataset	ACC	NMI	ARI
AG News	0.821 ± 0.068	0.670 ± 0.033	0.677 ± 0.067
TREC coarse	0.912 ± 0.070	0.892 ± 0.057	0.882 ± 0.075
DBpedia	0.986 ± 0.002	0.966 ± 0.003	0.969 ± 0.003

Table 3: Mean results $\pm$ std. deviations over 5 repetitions for overclustering with $n_{out} = 30$. The model performs well despite the absence of the true K.

K potential clusters. From the results in Table 3, we observe that CC copes very well with this challenging scenario. This motivates the extension towards DTD.

Following Section 3.3, we train five *Phase 1* models $f_{i,t=1}$ on $D_{train,t=1}$ and evaluate their performance on the three different test sets using the DBPedia data set. We use the KCL loss due to its superior performance in the previous experiments. We observe a decent performance on $D_{test,1}$ along with a correctly detected number of classes in Table 4. Note, that we consider a class as 'detected' if the model assigns at least one percent of the respective test set to the specific cluster. We acknowledge that this is a rather heuristic choice. For $D_{test,2}$ and $D_{test,combined}$, the models perform substantially worse and are not able to detect the correct number of classes. Still, it is noteworthy, that the model is able to detect that the four novel classes in $D_{test,2}$ are distinct as it assigns them different clusters and does not simply assign them one 'outlier' cluster. From the observation that the model detects a total of ten clusters, as opposed to the correct $K = 14$ for $D_{test,combined}$, we infer that while it realizes these four new clusters are distinct, it assigns them to the clusters present in $D_{train,t=1}$. However, the *Phase 2* model $f_{t=2}$ obtained by fine-tuning the best performing *Phase 1* for 200 epochs on $10,000$ constraints sampled from $D_{train,t=2}$ (50% new vs. 50% old) performs very well on all three test sets and is able to detect the correct overall number of classes. Refer to the confusion matrices in Figure 2 for further illustration of these results. When $D_{train,t=2}$ contains more samples from the 'old' classes (25% new vs. 75% old), overall model performance still improves compared to *Phase 1*, but substantially less compared to when there is more information about the 'new' classes. These results imply that the algorithm shows considerable sensitivity to the degree of novelty present in the new training data, which has to be investigated further in future research. This experiment shows how an OC-KCL model can easily be adapted to a dynamically changing number of clusters via continued training on pairwise annotations from newly incoming training data.

5 Discussion and Conclusion

In this work, we connected two branches of research: contemporary NLP research and weakly supervised learning approaches. While the usefulness of CC-KCL (and MCL) had already been shown for computer vision settings (Hsu and Kira, 2016; Hsu et al., 2019), we extended it towards NLP. Based on this, we showcased how existing shortcomings of ordinary supervised approaches – the requirement of fixed, static label sets – could be regarded as a new type of learning task which we introduced as *dynamic topic discovery*. Within DTD, we subsume a dynamic setting where an initial, weakly annotated training data set at time $t = 1$ is accompanied by a second data set at time $t = 2$ which contains novel classes unseen at $t = 1$. We proposed a potential solution for such DTD settings via an alternative training scheme leveraging the overclustering and category detection capabilities of CC models. We acknowledge that there are still numerous unsolved problems such as the application on *very* short texts, *very* large label sets with large class cardinality, or multi-label scenarios. Nevertheless, we hope that our experimental results can serve as a foundation for further research toward tackling these increasingly complex problems to ultimately reduce manual labeling efforts in NLP.

	Test set	ACC	NMI	Predicted K
Phase 1	$D_{test,1}$	**0.988** / 0.982 ± 0.009	**0.969** / 0.964 ± 0.005	10 (Range: $[10-10]$)
(**Best** / Mean $\pm$ Std. Dev)	$D_{test,2}$	**0.616** / 0.570 ± 0.043	**0.409** / 0.410 ± 0.048	4 (Range: $[4-5]$)
	$D_{test,combined}$	**0.717** / 0.710 ± 0.011	**0.809** / 0.808 ± 0.015	10 (Range: $[10-11]$)

		50% new – 50% old			25% new – 75% old		
		ACC	NMI	Predicted K	ACC	NMI	Predicted K
Phase 2	$D_{test,1}$	0.980	0.951	10	0.880	0.895	9
	$D_{test,2}$	0.971	0.929	4	0.951	0.866	4
	$D_{test,combined}$	0.978	0.953	14	0.832	0.887	12

Table 4: DTD (with KCL) on DBpedia for different ratios of new versus old classes in $D_{train,t=2}$, from which we sample the 10,000 constraints for Phase 2, controlling the degree of novelty. Phase 1 is based on five different models on $D_{train,t=1}$. For Phase 2, we pick the best Phase 1 model and continue training on the constraints from $D_{train,t=2}$ (no standard deviations, since no random initialization of any model weights for Phase 2).

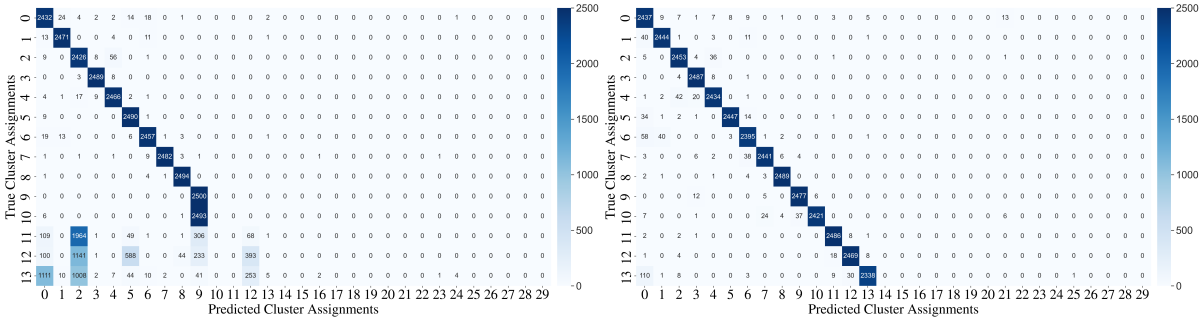

Figure 2: Confusion matrices for the two DTD phases on the $D_{test,combined}$. Phase 2 results (right) from the 50% new - 50% old setting illustrate a clear improvement over the results from Phase 1 (left). This shows that the Phase 2 model is able to cluster both the new and old data correctly.

Further, we believe that there is a high necessity for investigating DTD more in-depth. We believe it is important to design appropriate benchmarks and to investigate their relations to other dynamic paradigms, such as e.g. online learning or novel category discovery, and we hope this work can serve as a step in that direction.

Limitations

While we hope that this work provides valuable insights, there are still a couple of issues we did not yet address. First, we observed considerable instability during model training, especially for a lower number of constraints. Second, we found KCL to work better for DBpedia than MCL, which is surprising given the findings of Hsu et al. (2019). Finally, we (i) only evaluated DTD for one fixed set of constraints, (ii) only used the DBPedia dataset (due to the low number of classes in the other two datasets), and (iii) used a rather heuristic rule for determining the number of detected classes.

Acknowledgements

This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of BERD@NFDI - grant number 460037581. This work was supported by the Bavarian Ministry of Economic Affairs, Regional Development and Energy through the Center for Analytics – Data – Applications (ADACenter) within the framework of BAYERN DIGITAL II (20-3410-2-9-8).

References

David Arthur and Sergei Vassilvitskii. 2006. How slow is the k-means method? In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 144–153.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

Sugato Basu, Arindam Banerjee, and Raymond J Mooney. 2004. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004*

SIAM international conference on data mining, pages 333–344. SIAM.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

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.

Pierre Gançarski, Thi-Bich-Hanh Dao, Bruno Crémilleux, Germain Forestier, and Thomas Lampert. 2020. Constrained clustering: Current and new trends. In *A Guided Tour of Artificial Intelligence Research*, pages 447–484. Springer.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Yen-Chang Hsu and Zsolt Kira. 2015. Neural network-based clustering using pairwise constraints. *arXiv preprint arXiv:1511.06321*.

Yen-Chang Hsu and Zsolt Kira. 2016. Neural network-based clustering using pairwise constraints. *ICLR Workshop*.

Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. 2018. Learning to cluster in order to transfer across domains and tasks. *ICLR*.

Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. 2019. Multi-class classification without multi-class labels. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Shaohan Huang, Furu Wei, Lei Cui, Xingxing Zhang, and Ming Zhou. 2020. Unsupervised fine-tuning for text clustering. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5530–5534.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8360–8367.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *International Conference on Learning Representations*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Workshops*, Long Beach, CA, USA.

Tim Schopf, Daniel Braun, and Florian Matthes. 2021. Lbl2vec: An embedding-based approach for unsupervised document retrieval on predefined topics. In *WEBIST*, pages 124–132.

Douglas Steinley. 2004. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386.

Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Kiri Wagstaff and Claire Cardie. 2000. Clustering with instance-level constraints. *AAAI/IAAI*, 1097:577–584.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Hanlei Zhang, Hua Xu, Ting-En Lin, and Rui Lyu. 2021a. Discovering new intents with deep aligned clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14365–14373.

Hongjing Zhang, Tianyang Zhan, Sugato Basu, and Ian Davidson. 2021b. A framework for deep constrained clustering. *Data Mining and Knowledge Discovery*, 35(2):593–620.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Appendix

A Training Model Configurations

In Table 5 we list the specifications of the BERT-based language model that we use as architectural backbone which we obtained via huggingface (Wolf et al., 2020). We implemented our models and data loading logic in PyTorch (Paszke et al., 2017). Model training for the constrained clustering and the overclustering experiments was done on an NVIDIA A100-SXM4-40GB GPU with a batch size of 256 for 200 epochs. The models for the DTD part were trained on an NVIDIA Tesla-V100-16GB GPU with a batch size of 196 for 100 training epochs for phase 1 and for 200 training epochs for phase 2.

Parameter	Value
Base model	BERT-BASE-UNCASED
Learning rate	1×10^{-5}
Optimizer	AdamW (Loshchilov and Hutter, 2019)
Adam Epsilon	1×10^{-8}

Table 5: BERT configurations for all experiments.

B Detailed Results

In Table 6, we show results for the constrained clustering experiments with $n_{out} = K$ and a total of 20.000 constraint annotations for model training for the three datasets. This table includes mean $\pm$ standard deviations for the performance metrics across 5 repeated training runs to account for randomness in the training process. The results show that constrained clustering offers a viable alternative to supervised learning, almost reaching the upper baseline performance for the three datasets. Further, the MCL loss works best for the AGNews and the TREC-coarse datasets whereas the KCL loss is more suitable for the DBPedia dataset. Hence, we used the KCL loss in the experiments on DBPedia for the dynamic topic discovery experiments in Section 3.3.

C Performance metrics

Normalized Mutual Information (NMI) NMI is generally used to measure the tightness of the cluster formations. In other words, it quantifies if all the clusters are mutually exclusive without outliers (Strehl and Ghosh, 2002). Mathematically,

		Lower Baseline		
Data set	K	ACC	NMI	ARI
AG News	4	0.830	0.577	0.605
TREC-coarse	6	0.542	0.299	0.302
DBPedia	14	0.631	0.726	0.494
		CC-KCL		
AG News	4	0.870 ± 0.088	0.714 ± 0.059	0.739 ± 0.087
TREC-coarse	6	0.953 ± 0.007	0.890 ± 0.010	0.900 ± 0.012
DBPedia	14	**0.982** ± 0.005	**0.963** ± 0.005	**0.967** ± 0.009
		CC-MCL		
AG News	4	**0.917** ± 0.003	**0.755** ± 0.004	**0.795** ± 0.006
TREC-coarse	6	**0.967** ± 0.004	**0.908** ± 0.009	**0.923** ± 0.009
DBPedia	14	0.661 ± 0.057	0.805 ± 0.038	0.653 ± 0.055
		Upper Baseline		
AG News	4	0.919 ± 0.001	0.759 ± 0.005	0.800 ± 0.003
TREC-coarse	6	0.962 ± 0.002	0.897 ± 0.006	0.917 ± 0.005
DBPedia	14	0.989 ± 0.001	0.974 ± 0.001	0.977 ± 0.001

Table 6: Results for the baselines on all available training samples for all of the analyzed data sets as well as for CC-MCL and CC-KCL on 20,000 constraints each. The better CC model (between KCL and MCL) is marked in bold. Mean and standard deviations of the metrics over five runs.

NMI describes the change in entropy of class labels given the true cluster labels:

$$NMI = \frac{2 \cdot I(Y, \hat{Y})}{H(Y) + H(\hat{Y})}$$

where $I(Y, \hat{Y}) = H(Y) - H(Y|\hat{Y})$ is the mutual information. $H(Y)$ and $H(\hat{Y})$ are the entropy of the ground truth class label Y distribution and the entropy of the predicted cluster label distribution $\hat{Y}$, respectively. The NMI is bound to $[0, 1]$ where a higher score implies better clustering performance.

Accuracy (ACC) Accuracy measures the similarity of predicted results with the respective ground truth. For clustering accuracy, we use the Hungarian algorithm (Kuhn, 1955) to assign predicted clusters with associated class labels. Given ground truth classes Y and predicted clusters $\hat{Y}$ we calculate accuracy as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

Adjusted Rand Index (ARI) The ARI is used to measure the similarity between two clustering outputs (Steinley, 2004). Here, the actual class labels

are compared to predicted cluster labels to measure the clustering performance. When comparing Y and $\hat{Y}$, the ARI is calculated as follows:

$$R = \frac{a+b}{\binom{n}{2}}$$

where a is the number of times, pairs of elements are in the same cluster for Y and $\hat{Y}$, b is the number of times a pair of elements is not in the same cluster for Y and $\hat{Y}$ and n is the total number of samples in the batch.