IITK at SemEval-2021 Task 10: Source-Free Unsupervised Domain Adaptation using Class Prototypes

Harshit Kumar* **Jinang Shah*** **Nidhi Hegde***
Priyanshu Gupta* **Vaibhav Jindal** *
Ashutosh Modi
Indian Institute of Technology Kanpur (IIT Kanpur)
{hakumar, sjinang, nidhih, guptap, vaibhavj}@iitk.ac.in
ashutoshm@cse.iitk.ac.in

Abstract

Recent progress in deep learning has primarily been fueled by the availability of large amounts of annotated data that is obtained from highly expensive manual annotating processes. To tackle this issue of availability of annotated data, a lot of research has been done on unsupervised domain adaptation that tries to generate systems for an unlabelled target domain data, given labelled source domain data. However, the availability of annotated or labelled source domain dataset can't always be guaranteed because of data-privacy issues. This is especially the case with medical data, as it may contain sensitive information of the patients. Source-free domain adaptation (SFDA) aims to resolve this issue by using models trained on the source data instead of using the original annotated source data. In this work, we try to build SFDA systems for semantic processing by specifically focusing on the negation detection subtask of the SemEval 2021 Task 10. We propose two approaches - *ProtoAUG* and *Adapt-ProtoAUG* that use the idea of self-entropy to choose reliable and high confidence samples, which are then used for data augmentation and subsequent training of the models. Our methods report an improvement of up to 7% in F1 score over the baseline for the Negation Detection subtask.

1 Introduction

The availability of large scale datasets has been the main driving factor behind the success of supervised deep learning in the recent times. However, the process of data annotation is very expensive and time consuming, being one of the major challenges in extending deep learning techniques for new tasks.

One possible way to solve this problem is to *train* a machine learning model using an annotated source dataset to assist the annotation process over some unlabelled target dataset. However, there may be differences in the source and target domain distributions, which may lead to inaccuracies. Thus the challenge is to update the weights of the source classifier to generalize it well on the target domain. This aligns with the well studied problem of *Unsupervised Domain Adaptation* (UDA) (Kouw and Loog, 2019; Wang and Deng, 2018; Ramponi and Plank, 2020)

A common denominator across many popular *UDA* methods is their dependence on large amounts of labelled source domain data (Ganin and Lempitsky, 2015; Saito et al., 2018). However, many a times it is not possible to release the source domain dataset because of privacy concerns. This problem becomes particularly relevant when working with clinical Natural Language Processing (NLP) datasets because they contain highly sensitive information which cannot be freely distributed. To tackle these data sharing constraints, the framework of *Source-Free Domain Adaptation (SFDA)* is gaining interest (Laparra et al., 2020). In SFDA, instead of sharing the source domain data, only a model that has been trained on the source domain data is shared. This model is then used for solving the original task for the unlabelled target domain.

SemEval 2021 Task 10 (Laparra et al., 2021) asks participants to develop SFDA models for two subtasks. The first subtask involves **Negation Detection**, where we are required to determine whether or not a clinical entity *(diseases, symptoms, etc.)* mentioned in a sentence is negated in the given context. The second subtask is of **Time Expression Recognition**, where the objective is to detect and label all time expressions mentioned in a given document. In this work we have focused on the negation subtask. For solving this subtask our strategy is to make use of high-confidence prototypes from the *target* domain to reinforce the *target-*

* Authors contributed equally to the work. Names in alphabetical order.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 438–444
Bangkok, Thailand (online), August 5–6, 2021.

specific features of the source model. We propose a simple augmentation technique that makes use of these *high-confidence* prototypes to generate labelled artificial datapoints. These augmented samples are then used to perform supervised fine-tuning of our source model. Using our methods, we were able to obtain upto a 7% improvement in F1 score over the baseline. The code for models and experiments is made available via GitHub.[1]

2 Background

In source free domain adaptation (SFDA) problem for semantic analysis, the goal is to create accurate systems for un-annotated target domain data. For these tasks, we are provided with un-annotated target domain data, and a model trained on the annotated source domain data for a similar task.

The shared task is further divided into 2 subtasks – Negation detection and Time expression recognition. In this work we focus only on the first subtask.

Negation Detection: The task is to classify clinical event mentions for whether they are negated by their context. This is essentially a "span-in-context" classification problem, where both the entity to be classified and its surrounding context are to considered. For example, the sentence - "*Has no <e>diarrhea <\e>and no new lumps or masses*" has the entity *diarrhea* which is negated by its context, and the model's task is to correctly identify this entity as negated.

Pretrained Models: For negation detection, a given pre-trained classification model has been fine-tuned on the 10,259 instances in the SHARP Seed dataset(Rea et al., 2012) of de-identified clinical notes from Mayo Clinic of which 902 instances are negated.

Practice Data: The development data for the negation task is a subset of the i2b2 2010 Challenge (Uzuner et al., 2011) on concepts, assertions, and relations in clinical text. The practice dataset is further divided into *train* and *dev* splits. The *train* split contains 2886 unlabeled sentences while the *dev* split is composed of 5545 labeled sentences.

Test Data: A part of the MIMIC III corpus v1.4 (Johnson et al., 2016),is used as the test set for the negation detection subtask. The processed test data contains around 600K instances.

[1] https://github.com/purug2000/protoAug.git

2.1 Prior Work

The limitations in creating large scale annotated datasets have led to a large amount of work on unsupervised domain adaptation in the recent years (Ganin and Lempitsky, 2015; Ganin et al., 2016; Tzeng et al., 2017; Saito et al., 2018). However, most of this work assumes free availability of source domain data. In source free domain adaptation (SFDA) problems, when no annotated source data is available, and only a pretrained model is provided, the domain adaptation problem becomes rather difficult, and this remains a largely unexplored area in the NLP community. However, there have been some recent works in the computer vision domain that attempt to solve this problem. Hou and Zheng (2020) propose a model to transfer the style of source images to that of the target images by exploiting the information stored in the batch normalization layers of the pre-trained model. In another work, (Kim et al., 2020) observed that the target domain data points with lower entropy are generally classified correctly and are reliable enough to generate pseudo labels for the entire target dataset.

The two sub-tracks for the current SemEval task are well studied problems in the supervised setting and a lot of work has been done on developing models for both the negation detection in clinical settings (Chapman et al., 2001; Cotik et al., 2016) and the time expression recognition taak (Laparra et al., 2018). However, in this work, we attempt to approach the negation detection task from the perspective of *SFDA*, and not on improving these techniques in general.

3 System Overview

In this paper we offer a novel perspective on the problem of domain adaptation for the negation detection task in clinical NLP. The proposed approaches attempt to utilize some of the aspects of both self-learning and semi-supervised learning, as explained next.

Class Prototypes: If there was any access to the labeled target data then the most intuitive approach would have been the fine-tuning. But for unlabeled case, in order to fine-tune the pre-trained network S, it would become necessary to generate a labeled set of data from the given unlabeled target domain data. One way to approach this would be through a concept from self-learning, i.e., by finding the most reliable samples from the target data over which the

model S is sufficiently confident and using these predictions as the corresponding ground truth. In order to find reliable target samples, self-entropy H can be used to quantify the prediction uncertainty:

$$H(x) = -\sum_{k=1}^{K} p_k^S(x) log(p_k^S(x)) \quad (1)$$

Here, S refers to the network (pre-trained classifier), K are total number of classes, and $p_k^S(x)$ is the probability of the instance x belonging to the class k.

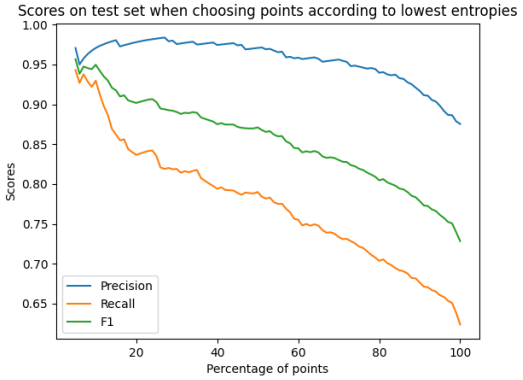

Figure 1: For any point P on the x-axis, the y-axis shows the performance scores on the data points having their self-entropies in the lowest P percentile.

The samples with smaller self-entropy indicate that the classifier is more confident over them, and these are referred to as *Prototypes*. The relationship between the self-entropy value and certainty in the prediction can be clearly observed in figure 1. Due to very high confidence of the model and accuracy over these prototypes, we can safely consider their predicted labels as their actual true labels. Here, one crucial hyperparameter to consider is the self-entropy threshold below which the residing target samples will be identified as prototypes. The key issue faced while determining the value of this hyperparameter is the disparities and highly imbalanced data distribution between the classes. Especially, in the negation task, the presence of the negated sentences in the practice data itself is in a very small proportion than that of non-negated sentences making negated a minority and non-negated a majority class. Analysis of the practice data (figure 2) further showed that the lowest self-entropy achieved by the negated class is far higher than that of non-negated class. For now, the self-entropy threshold value is defined by the 50th percentile (median) self-entropy value of the minority class i.e. negated class. The rationale for this is provided in the Analysis section (section 5).

Unfortunately, the fine-tuning of the pre-trained network S with the prototypes identified from the target samples, didn't enable the trained network to generalise its performance over unseen target data as it made it highly likely to overfit on the prototypes without any early intervention. With this approach, we were able to get improvement of about 1% in F1 score over the baseline but that too with the highly unstable and unreproducible training results. Only using the reliable samples became a major issue as the trained model seemed to be unaware of the samples over which the model was less confident before.

Augmentation: To address the issue of lack of generalisation of the model towards the unseen and not-so-reliable target data, we propose the use of **augmentation**, inspired by the Mean Teacher model proposed by Tarvainen and Valpola (2018). The basic idea is to regularize the network by introducing noise to the prototypes in a way that can keep the label same and thereby creating new data points with the known labels. Mean Teacher model uses similar strategy for the labeled data points, instead here we apply that to the identified prototypes from the target data. This way, the pre-trained network can be subjected to a further constructive training by introducing a set of new labelled samples, which could help with the generalisation for the trained model.

Another use of the augmentation here is to address the issue of highly skewed and imbalanced distribution of data between the classes. Here, the augmentation is utilized not just to generate new samples to regularize the network but also to make the data distribution balanced across the classes by adding new samples in accordance with the preferred proportions.

Although there could be many possible ways to augment the samples so that their labels can be maintained, for this specific task, since we have a highlighted concept term in each sentence, we have used the augmentation by replacing that concept term. The new sentences are generated from their parent prototype sentences by replacing the concept term with a concept term from any randomly selected sentence of the same class in the data set. Here we have assumed that most of the concept terms will represent medical condition and thus nouns. Even when the grammar and the sentence

structure is preserved, the result of the augmentation might still not be perfect due to the possible ambiguity in the concept term selection or incorrect grammar in the original sentence itself. For example, consider the sentence *"...<e> cortisol </e> is currently pending."*. Here it is not clear if the entity "*cortisol*" is negated or non-negated. However, it should be noted that such ambiguities are relatively less frequent which in turn implies that the augmentation will suffice its purpose for the majority of the time.

ProtoAUG and Adapt-ProtoAUG: Combining the concepts discussed above, here we propose our two approaches *ProtoAUG* and *Adapt-ProtoAUG*, both of which share the same set of concepts of class prototypes and augmentation as explained before. For both ProtoAUG and Adapt-ProtoAUG, the common underlying procedure is as follows: first the prototypes are identified from the target domain data and then the augmentation follows with the intent of regularizing the network and to create a more balanced distribution of samples across the classes. Now both the prototypes and their augmented samples with their respective labels are used with cross entropy loss to update the weights of the feature extractor module F of the pre-trained network S, with classifier module C of the pre-trained network being frozen.

The fundamental difference between ProtoAUG and Adapt-ProtoAUG is about the adaptive nature of Adapt-ProtoAUG in recognizing the prototypes from the original target domain dataset after every epoch, which ProtoAUG does only at the beginning of the training. Adapt-ProtoAUG makes incremental changes to the percentile score (initially 50) for the self-entropy threshold. The intuition behind using this strategy is that as the training proceeds, model will become more confident on the training samples and the entropy values for all the samples will significantly decrease. A possible drawback of using a fixed percentile criteria for the threshold at every epoch would likely exclude some reliable samples even when they achieve objectively quite lower values of self-entropy. To avoid such scenarios, apart from repeating the same process of prototype identification followed by augmentation after every epoch, we also propose to increase the percentile score for determining the self-entropy threshold in an uniform manner throughout the training with some fixed upper bound (70). This is also explained further in the Analysis (section 5).

4 Experimental Setup

Model: The pre-trained model used in subtask-1 is a RoBERTa (Liu et al., 2019) based sequence classification model[2] provided by organizers after training on source domain data inaccessible to us. It has two modules - a feature extractor and a classifier that operates over the output [CLS] token from the feature extractor.

Data: In the practice phase, *train* split of the practice dataset was used to further train the pre-trained model whereas the *dev* split was used as a validation set for the hyper-parameter tuning. In the evaluation phase, due to our computational constraints, we were only able to utilize a randomly selected subset of 25k samples from around 600k sentences of the test dataset, for retraining of the pre-trained model. For evaluation the organisers use an annotated subset of the test dataset. During the evaluation phase this subset was kept hidden from the participants.

Hyper-parameters setting: For ProtoAug, self-entropy percentile threshold is set to 50% whereas as for Adapt-ProtoAUG it is uniformly increased after every epoch from being 50% at the first to being 70% at the final epoch. Using augmentation, the final number of samples per class is set to be x times the number of prototypes belonging to the majority (non-negated) class. In our experiments, we choose x to be 4. For both the approaches, the model training is performed for 10 epochs. During the test phase, we reuse the hyper-parameters obtained from the practice phase. Further details about hyper-parameter selection can be found in Appendix A.

5 Results

Table 1 shows results on the development data for the two approaches - ProtoAUG and Adapt-ProtoAUG. For reference, results obtained from the pre-trained model are shown as the baseline.

Model	F1 score	Precision	Recall
Baseline	0.834	0.850	0.818
ProtoAUG	0.877	0.948	0.816
Adapt-ProtoAUG	**0.888**	**0.959**	**0.827**

Table 1: Results obtained on dev data of practice phase

[2]Model is available on https://huggingface.co/tmills/roberta_sfda_sharpseed

Table 2 shows results of our two proposed approaches with the baseline model on the test set. In evaluation phase, the observed improvement in F1-score was of roughly 7% from the original baseline.

Model	F1 score	Precision	Recall
Baseline	0.66	0.917	0.516
ProtoAUG	0.706	**0.939**	0.566
Adapt-ProtoAUG	**0.729**	0.876	**0.624**

Table 2: Results obtained on test data of evaluation phase

Analysis: To ascertain the relationship between self-entropy and a prediction's reliability, we analyse the baseline performance scores for the data points within the varying self-entropy percentile threshold as shown in figure 1. For the baseline, we observe a direct correlation between a lower self-entropy and a higher prediction score on the respective data points. This further supports the use of low self-entropy data points as class prototypes in our proposed approaches.

As shown in figure 2, the lowest self-entropy achieved by minority (negated; label 1) class is far higher than that of majority (non-negated; label -1) class. This may be attributed to the skewed nature of the target dataset and potentially the source dataset as well.

Another interesting observation from figure 2 is that most of the majority class samples have a self-entropy lower than the lowest self-entropy achieved by the minority class. Thus, for selecting the threshold for prototype selection, we apply the percentile-based criteria only on the self-entropy values of the minority class.

For prototype selection, instead of using an absolute threshold value, we have chosen a percentile-based entropy threshold as it adapts relatively well across different domains. This follows from the fact that confidence of the model may vary from domain to domain due to which a threshold chosen for one domain might not be a good criteria for another domain.

For Adapt-ProtoAUG, as the upper bound for the self-entropy threshold is increased beyond 70, we observed a gradual decline in the model's performance for the dev set. This may be due to the 85% precision of the baseline. Precision here refers to the proportion of correctly classified negated samples to the model's total number of negated predictions. So, it could be the case that as the threshold reaches near or get past the precision score, the probability of identifying a wrongly labelled sample as a prototype will rapidly increase. Compared to ProtoAUG, as the percentile threshold was increased from 50 to 70, we observed an overall increment of recall for both the dev and test dataset. Furthermore, introducing augmentation in the framework drastically increased the stability and reproducibility of the training process.

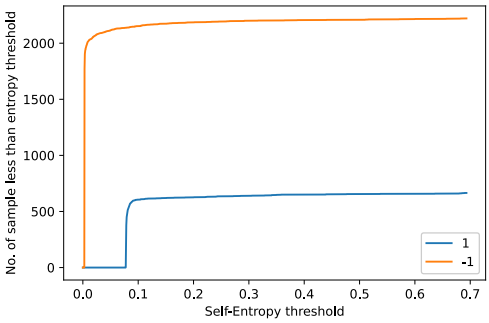

Figure 2: Cumulative number of sample vs entropy threshold curve

Appendix B provides some insights from t-SNE analysis, visually justifying the performance improvement. It analyses the predictions of the baseline model and Adapt-ProtoAUG over a fixed two-dimensional feature-space, comparing the similarities of their respective predictions with the original ground-truth labels. We observe that Adapt-ProtoAUG can capture the label distribution better than the baselines by performing well on various non-trivial data-points.

6 Conclusion

In this work, we carefully explored the problem of source-free domain adaptation for the Negation Detection subtask. We studied the importance of the confidence that a model places on its prediction and analyzed its formulation in terms of the samples' self-entropy scores. Further, using those insights, we proposed two simple and intuitive approaches, namely ProtoAUG and Adapt-ProtoAUG for the Negation Detection Subtask and got an improvement of 7% on the test set with respect to the baseline model.

References

Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34(5):301 – 310.

Viviana Cotik, Roland Roller, Feiyu Xu, Hans Uszkoreit, Klemens Budde, and Danilo Schmidt. 2016. Negation detection in clinical reports written in German. In *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)*, pages 115–124, Osaka, Japan. The COLING 2016 Organizing Committee.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France. PMLR.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.

Yunzhong Hou and Liang Zheng. 2020. Source free domain adaptation with image translation.

Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1).

Youngeun Kim, Sungeun Hong, Donghyeon Cho, Hyoungseob Park, and Priyadarshini Panda. 2020. Domain adaptation without source data.

Wouter M. Kouw and Marco Loog. 2019. An introduction to domain adaptation and transfer learning.

Egoitz Laparra, Steven Bethard, and Timothy A Miller. 2020. Rethinking domain adaptation for machine learning over clinical language. *JAMIA Open*, 3(2):146–150.

Egoitz Laparra, Dongfang Xu, Ahmed Elsayed, Steven Bethard, and Martha Palmer. 2018. SemEval 2018 task 6: Parsing time normalizations. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 88–96, New Orleans, Louisiana. Association for Computational Linguistics.

Egoitz Laparra, Yiyun Zhao, Steven Bethard, and zlem Uzuner. 2021. SemEval 2021 Task 10 - Source-Free Domain Adaptation for Semantic Processing(to appear). *Proceedings of the 15th International Workshop on Semantic Evaluation.*

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey.

Susan Rea, Jyotishman Pathak, Guergana Savova, Thomas A. Oniki, Les Westberg, Calvin E. Beebe, Cui Tao, Craig G. Parker, Peter J. Haug, Stanley M. Huff, and Christopher G. Chute. 2012. Building a robust, scalable and standards-driven infrastructure for secondary use of ehr data: The sharpn project. *Journal of Biomedical Informatics*, 45(4):763–771. Translating Standards into Practice: Experiences and Lessons Learned in Biomedicine and Health Care.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Antti Tarvainen and Harri Valpola. 2018. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

zlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Mei Wang and Weihong Deng. 2018. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153.

A Training Hyper-parameters

We train our models with a batch size of 32 using SGD optimizer with initial $lr = 0.0005$ which decays after every iteration by multiplying it with $(1 + 10 * itr/max\_iter)^{-0.75}$ (where itr is current iteration and max_iter is maximum iteration of training), $weight\_decay = 0.0005$ and $momentum = 0.9$.

B t-SNE Analysis

We performed low dimensional analysis of the models using tsne. In the following figures, we took the 768 dimensional output of the baseline roberta model for the test dataset, and projected it in two dimensions using tsne.

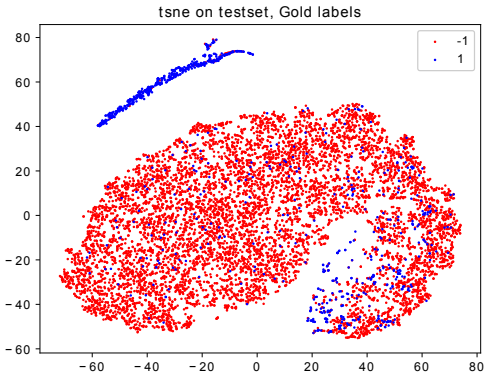

Figure 3: Ground truth labels. Blue points correspond to negation = 1.

- Figure 3 shows the the 2-dimensional tsne plots with the original ground truth labels.The blue points are the true positives and reds are the true negative points. We broadly observe two clusters - a smaller one with majority of points being true positives and a larger cluster with a majority of points being negatives. We also observe some blue points scattered in the red cluster and vice-versa.

- Figure 4 shows the predictions of the baseline model on the target domain. We see that the baseline classifier segregates the test data into almost perfect clusters, and thus misclassifies the scattered points.

- Figure 5 shows the prediction results of adapt-protoaug. In this case the F1 score improved from the baseline score by around 7%. Looking at the figure, we clearly see an improvement with respect to the baseline model, as we are now able to correctly capture some of the points that randomly fall within in the opposite cluster.

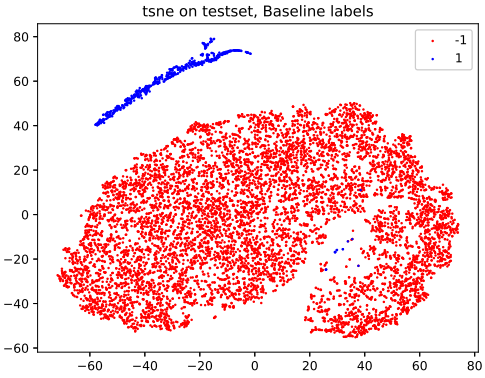

Figure 4: The predictions of the baseline model on the test set of the target domain.

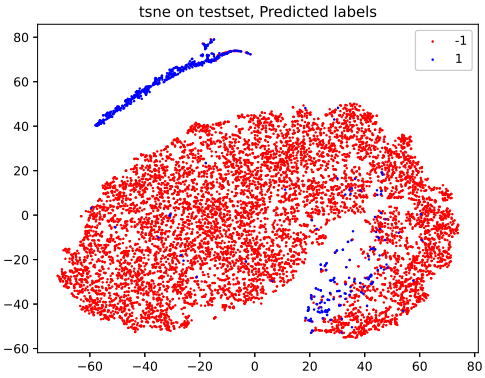

Figure 5: Visualisation of the predictions of an improved model