# Adaptive sampling for multimodal anomaly detection Project Report

Ziyi Liu
University of Cambridge
zl413@cam.ac.uk

***Abstract*—We adapted our own multimodal classification model based on OSCAR model that aligns images and texts with object tags for hate speech detection task. In addition to raw text-image paired input, the improved model was tailored to further include high-level features as auxiliary inputs, such as correlation between different sections of texts and sentiment analysis. We then proposed a systematic approach of performing model-agnostic active sampling on multimodal classification tasks with various uncertainty estimation methods.**

## I. Introduction

Text classification task in social media for flagging inappropriate speeches is particularly challenging due to the high level of noise and lack of context in the data collected. Text channel on its own often misses important connotations and subtexts complemented by the accompanying images, so that a language-only classification model would miss out such critical information for inference.

Recent studies have explored the Vision+Language representation learning that aligns image and text data for better performance on downstream tasks, such as Question Answering and Video Captioning. Many of these methods () were based on Transformer models (VisualBERT [1], VL-BERT [2], ViL-BERT [3], LXMERT [4], UNITER [5], etc.), but with different alignment and fusion techniques. Some other methods involve extracting textual information from the image modality before using a multimodal approach in hate speech detection [6]. Given our imbalanced binary classification task with very noisy (and sometimes missing) data pairs, we picked OSCAR [7] as the base model for several reasons:

1) object tags are extracted as anchors between word tokens and image region features
2) flexible input processing when one or more modalities are missing
3) state of the arts performance on multiple downstream tasks (VQA, NLVR2, GQA, etc.)

Despite the improved performance of transformer-based multimodal models compared with RNN-based ones [8], they are highly dependent on a multitude of low-noise data for training and/or fine-tuning. In real world applications, training on a large set of noisy data is not only costly in time, but the inconsistency of data quality may also confuse the model, which results in ineffective representation learning. We approach this problem with an active learning approach, where we find the most uncertain data points iteratively for each training round, and we train the model with these data only to maximize the training efficiency [9].

## II. Proposed Model

The base OSCAR model (Fig. 1) is an improvement of many recent methods due to its more involved input representations and a different pre-training objective. We denote the input triplet as ($\mathbf{w}$, $\mathbf{q}$, $\mathbf{v}$), representing text embeddings, object tags embeddings, and image region vectors. As indicated from fig. 1, we may interpret the triplet from two different perspectives - dictionary view and modality view - where Masked Token Loss and Contrastive Loss are used in each interpretation, respectively. The final pretraining loss of OSCAR is then

$$\mathcal{L}_{pretraining} = \mathcal{L}_{MTL} + \mathcal{L}_{C}$$

.

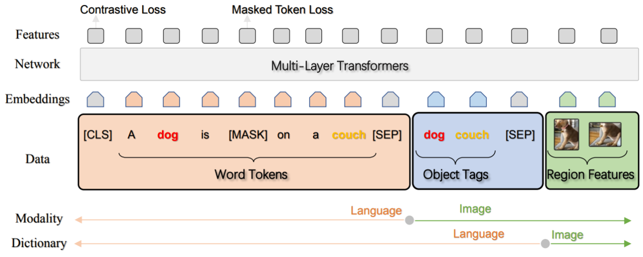


Fig. 1. Diagram of OSCAR Model, where the (language, object-tag, image) input pair is illustrated with a Transformer-based model.

Our own model takes on the triplet-input method with a number of additional features and innovations tailored for our classification task and dataset. The main changes and features are enumerated below:

- Correlation score between title and main text(see implementation at this commit): The degree of similarity between the title and the main text was calculated with their cosine distance from their text embeddings generated by pre-trained BERT model. The intuition is that the higher degree of dissimilarity, the more likely that the texts contain hate speech due to the higher likelihood of ’click bait’ and a greater inconsistency.
- Sentiment scores of main text body (see implementation at this commit): The sentiment of each piece of text is extracted with TextBlob, which was chosen for its simplicity and very fast inference time compared with transformer-based sentiment analyzers. Two key results are outputted and then in turn used for our training as features: polarity and subjectivity. Polarity, in the range of [-1,1], measures how negative/positive the text

is, which helps capture the negativity in hate speeches, while subjectivity, in [0,1], can help pinpoint the strongly opinionated and subjectively biased passages.

- Replacing 0-1 labels with confidence score: In some of our datasets, entries were labelled by three independent assessors, and therefore the final label can be averaged to be selected from {0, 0.33, 0.67, 1}, as opposed to a hard 0-1 label.

Upon obtaining the features from the multi-layer transformer model, we aggregate our own extracted features with the transformer outputs, and pass them into a classifier head for hate speech classification downstream task. Given a pretrained OSCAR base model, we fine-tune the complete model with a Cross Entropy loss ($\mathcal{L}_{CE}$).

The modified classifier model can be illustrated in fig. 2 on a high level.

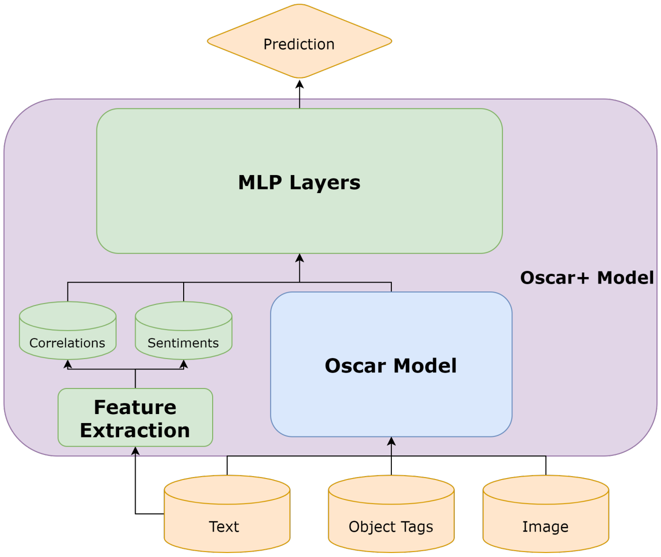


Fig. 2. High level diagram of modified OSCAR model (OSCAR+)

## III. Active Learning Setup

The goal of employing active learning is that if we could attain a comparable level of test set accuracy, if not higher, with a considerably smaller set of instances, then we could terminate the training procedure without including the rest of the data. This is particularly useful when we have a large pool of unlabelled data, so that we would immensely reduce the human labelling efforts before the training procedure.

Having established and implemented our improved model, we then treat it as a black box module that can be directly integrated with the active learning algorithm. Based on the original implementation from this codebase, we implemented an adapted version with additional features for our hate speech datasets (see the relevant branch here). We adopted a standard Bayesian active learning approach, and the procedure can be illustrated in fig. 3. Specifically, we executed the following procedure:

1) Randomly label a small number of samples in the unlabelled pool to initialize the procedure
2) For some heuristic h, uncertainties are calculated for all unlabelled data in the pool
3) Top n samples in the pool with highest uncertainties are selected and labelled
4) Repeat steps 2-3 until the pool is empty

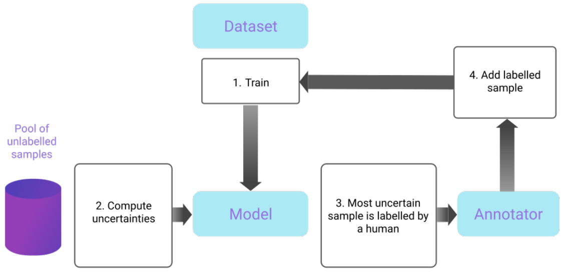


Fig. 3. Illustration of our active learning procedure [10]

The two main branches of methods for evaluating prediction uncertainty are 1) Monte Carlo Dropout estimates [11] and 2) Ensemble methods. The essential idea behind both approaches is having an ensemble of non-deterministic models and making $m$ predictions for each data point, so that each data entry has a result vector of length $m$. This is then used for uncertainty estimation with different measures. We experimented with a variety of these methods, including BatchBald [12], Mutual Information, Variation Ratio, and Max Entropy.

Furthermore, we came up with a variation of the active learning setup above, where we use model checkpoints. In other words, in each active learning iteration, we select $n$ instances for the model from the last checkpoint to train on, instead of keeping all cumulated samples from previous iterations. The implementation can be seen from this commit

## IV. Datasets and Related Work

For both model training and active learning, we used two main sources of data: MMHS150K [8] and our own Reddit data. The former contains 150,000 manually annotated tweets with text-image pair, where 3 human annotators classified each instance into 6 categories: Not hate, racist, sexist, homophobic, religion-based attacks, or attacks against other communities. We then aggregate all tweets that are not labelled ’non-hate’ as hate speeches, and the overall hate-normal split is 18.3% against 81.7%. We can clearly observe a class imbalance in the dataset, and this is taken into account with our optimiser choice of weighted Adam algorithm.

The latter dataset was scraped from Reddit, containing only 1466 instances in total, in which 500 are for testing. Intuitively, the difference between hate and non-hate in this dataset is more nuanced, and the class imbalance is much smaller than that of MMHS150K.

Before aligning with text embeddings, object tags and the image region features were extracted with Faster R-CNN [13] algorithm, where we leveraged the implementation here.

Note that this step is done separately from the main training procedure, such that the image features and object tags were pre-prepared with the texts.

In many images from both datasets, we can often see memes, pictures with text comments, or images of texts and books in real life. In these scenarios, object detection and regional features couldn't convey much useful information. Therefore, the OCR functionality is also included in our codebase. However, since we are passing in the paired inputs without raw images, this feature is left out for our experiments, but one could easily perform OCR as a pre-processing step. The OCR library can be found here

## V. Experiments and Results

### A. Training configuration

*Model correctness:* At the start of experiments, model training repeatedly resulted in the model predicting the same class for all samples. In order to demonstrate the model correctness, we trained the model on only one batch for 20 epochs, and the loss curve is plotted in fig. 4. Since we can observe that the model completely overfits, the model can indeed learn from the data we provide, albeit very slowly.

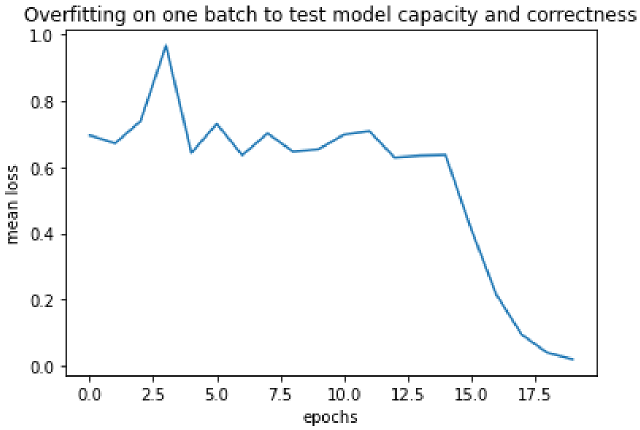


Fig. 4. Training loss for overfitting the model on one batch of data

We further show, in later experiments, the curves for training and validation loss are what we expected for a model with high enough capacity to learn (see fig. 5)

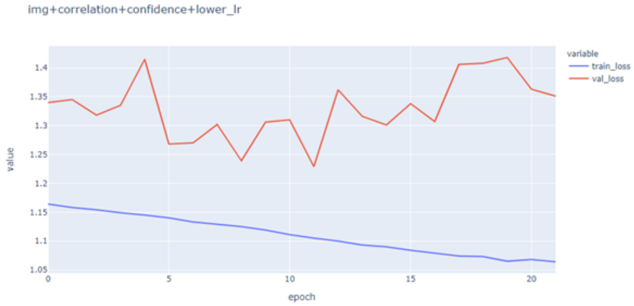


Fig. 5. Training and validation loss with the full dataset for correctness test

*Data Parallelism:* We experimented with data parallelism in order to accelerate our multi-gpu system. However, when we use more than one GPU, we could see the model outputting the same result for all data in both training and inference time. This issue goes away when we only use one GPU, but because of this, the effective maximum batch size was 8 at the maximum; otherwise, an ”out of memory” error would fatally crash the program. Therefore, in all our standard training procedures, we used one GPU with a small batch size of 4 or 8.

*Training setup:* In our standard training procedures, we chose our learning rate from the range [1e-7, 5e-5], where the best results were obtained with lr = 5e-6. We experimented with SGD, Adam, and AdamW, and most experiments were done with AdamW in the end given its better performance and easier implementation of weighted optimization accounting for class imbalance. Cosine learning rate scheduler was also utilized. The default epoch number was set at 20 with early stopping in place (where in our experiments the training typically halts before 15 epochs). Most of the other model-specific configurations were left unchanged from the OSCAR setting, Also, The object tags and image features were pre-processed before being fed into the model to save some time and memory during training.

### B. Model Performance

We first train our model alone on the larger MMHS150K dataset and evaluate on the test set for correctness. We also conducted ablation studies with different features, as shown in table I.

We also show our best result on the ROC curve in fig. 6 with the MMHS150K dataset. However, the results were not very satisfactory with the smaller reddit dataset, as shown in fig. 7. One main reason for this divergence is that the number of training samples is far from enough for the transformer-based model to capture the nuanced differences inherent in the dataset. Also, it can be attributed to the fact that we did not use the OSCAR model pretrained with its own losses.

We can compare these results with the previous SOTA stats, as shown in fig. 8. We concentrate on the ROC-AUC measure here for comparison, as accuracy poorly reflects the true performance due to the high degree of class imbalance. Our best AUC of 0.739 slightly outperforms the previously best of 0.734 with FCM model. This clearly demonstrates the efficacy of our own model, despite the relatively small improvement. One main reason is that we did not pre-train the model with OSCAR-losses but instead trained it directly on the Cross Entropy loss for classification. Also the dataset itself is very noisy with a multitude of irrelevant information and a delicate boundary between the two classes. For example, attacking a politician with the same message may not constitute as hate speech in this case. And there were a few instances where our own judgement differed from the true 'label' as well.

We further benchmarked our full standard model against the text-only and image-only baselines to see if the model benefits from multi-modality. Surprisingly, the ROC-AUC

TABLE I
TRAINING RESULTS FOR BASELINE VARIATIONS OF PROPOSED MODEL

| Model Variation | Input type(s) | Accuracy | ROC-AUC | F1 |
|---|---|---|---|---|
| $OSCAR+$ | T, IT, I, O, S, C | 0.686 | **0.738** | 0.706 |
| $OSCAR+$ w/ binary labels | T, IT, I, O | **0.692** | / | 0.680 |
| $OSCAR+$ w/ lr=1e-6 | T, IT, I, O | 0.686 | / | 0.705 |
| $OSCAR+$ w/ lr=5e-6 | T, IT, I, O | 0.677 | / | **0.707** |
| $Textonly$ Baseline | T | 0.677 | **0.738** | / |
| $OSCAR+$ w/o pre-training | T, IT, I, O | 0.681 | **0.738** | 0.702 |
| $ImageOnly$ Baseline | I | / | 0.589 | / |

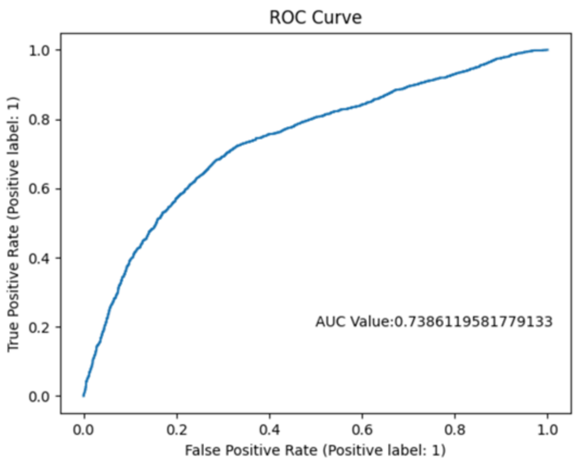


Fig. 6. ROC curve for the best-performing model setup on MMHS150K dataset

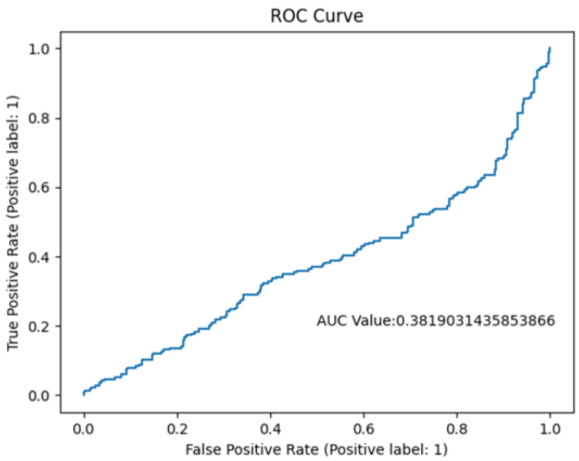
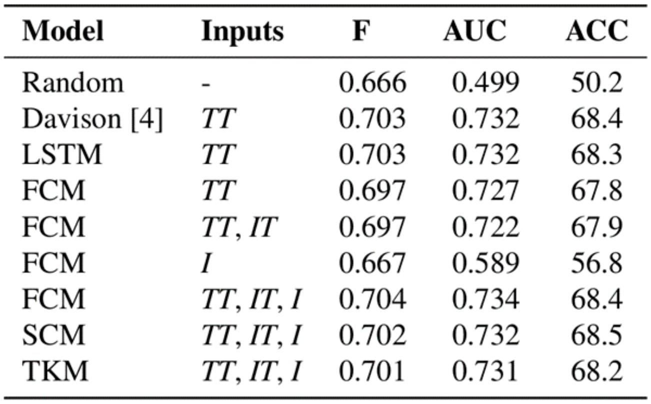


Fig. 7. ROC curve for our model trained on small Reddit dataset.

metric turned out identical as the full model for text-only case, although there's a considerable drop in test accuracy. On the other hand, the image-only model did not perform well as expected. Therefore, in this particular setup, the images and other stemmed high-level features may have only played peripheral roles in classification.



| Model | Inputs | F | AUC | ACC |
|---|---|---|---|---|
| Random | - | 0.666 | 0.499 | 50.2 |
| Davison [4] | *TT* | 0.703 | 0.732 | 68.4 |
| LSTM | *TT* | 0.703 | 0.732 | 68.3 |
| FCM | *TT* | 0.697 | 0.727 | 67.8 |
| FCM | *TT*, *IT* | 0.697 | 0.722 | 67.9 |
| FCM | *I* | 0.667 | 0.589 | 56.8 |
| FCM | *TT*, *IT*, *I* | 0.704 | 0.734 | 68.4 |
| SCM | *TT*, *IT*, *I* | 0.702 | 0.732 | 68.5 |
| TKM | *TT*, *IT*, *I* | 0.701 | 0.731 | 68.2 |

Fig. 8. State of the Arts performance for hate speech detection on MMHS150K with LSTM-based model.

### C. Active Learning

We use three main metrics for the full active learning procedure - accuracy, F1 score, and ROC-AUC - for our experiments. By default, we use the MMHS150K dataset for active learning processes. However, since each iteration of uncertainty evaluation and sample selection takes $\tilde{3}$ hours, we demonstrate some effects with a smaller subset of the full dataset (a randomly selected subset of 5000 samples).

Firstly, with the full dataset configuration, we compare the performance of different uncertainty measures, as shown in fig. 9. For running time considerations, we used a very large step size of 40,000 samples for each active iteration. All the other model-related hyperparameters in this and the following experiments are taken from the best-performing experiment in the model-training section. We see some erratic behaviour in some strategies (such as mutual information and expected entropy) due to the large step size, but the other three were as expected. However, we can see from this early-stage experiment that picking instances randomly can achieve the same level of or even better performance as other strategies.

For a clearer comparison, we single out the the most effective measure and benchmark it against the random baseline (fig. 10)

Following the preliminary experiments, we then verified the selection fairness to ensure that the instances selected at each step are not strongly biased from the mean (inspired by [14]). We mainly looked at the average text length and label distribution, as shown in fig. 11, 12. It can be concluded that

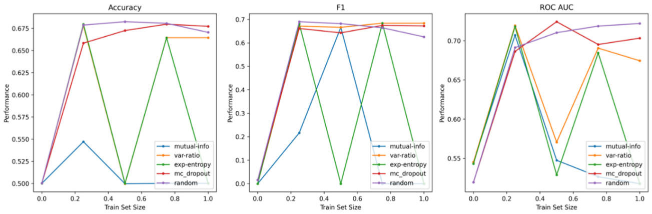


Fig. 9. Comparison of uncertainty estimation methods with a large step size of 40,000 samples in an active learning setup

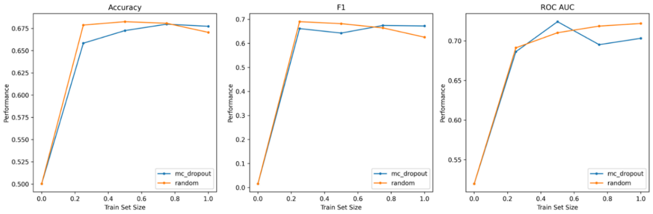


Fig. 10. benchmarking MC-Dropout uncertainty estimation method against the random baseline with a large step size of 40,000 samples in an active learning setup

no significant selection bias occurred in the active learning sampling process, as the fluctuations are within a reasonable range around the average.

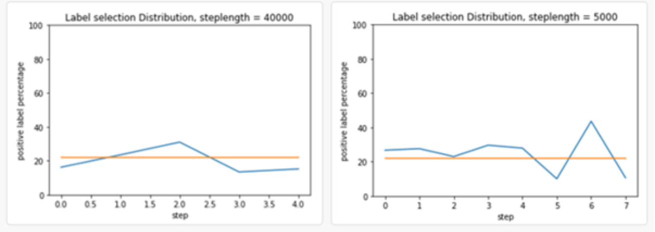


Fig. 11. Label distribution of samples selected in each active learning iteration(blue) contrasted with the average label distribution of the full dataset(orange).

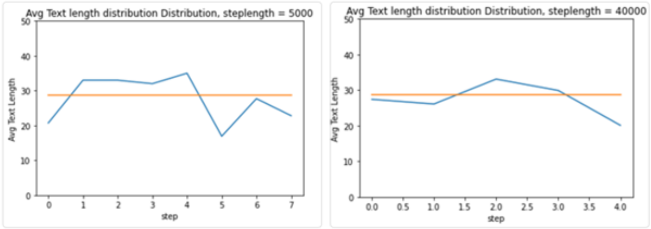


Fig. 12. Average text length distribution of samples selected in each active learning iteration(blue) contrasted with the average of the full dataset(orange).

We then adopted a lighter configuration described above, with a step size of 100 out of 5000 instances in total, to further investigate active learning in our multi-modal scenario when more steps are included. The full palette of plotted results are shown in the appendix.

An stark comparison between the experiments done with cumulative samples in the training pool and those with model checkpoints and only trained on one batch of samples each time can be observed. The performance of the latter appears to be stochastically varying around the ”random guess”, reflected from the ROC-AUC plots of several different uncertainty measures; whereas with the former approach, the random baseline almost paralleled most of the other methods in performance. However, one very interesting finding was with the variation ratio measure (see fig. 16), which attained a superior performance with only half of the training subset. This finding confirms our hypothesis that it is possible to select the instances deliberately to both improve the overall performance and reduce the number of samples needed.

We repeat some of the experiments with the full dataset using a step size of 5000 (see fig. 20, 21). We can see that in both cases, less than 20% of the total data is needed for the full-dataset performance. In fact, in the random situation, more instances after the first 20% leads to a gradually worse performance, while the max-entropy strategy kept the performance levelled.

Finally, we also show the change of estimated uncertainties stats with different strategies in figs 22, 23, and 24. We can see that different uncertainty estimation strategies have distinct profiles with respect to the selections steps, and therefore, picking the most suitable estimate according to the uncertainty curve is paramount for the most efficient active learning procedure. In this case, the mean variation ratio progression is what we expected: a growing uncertainty of data in pool as the model started to learn, followed by a gradual decrease as the model has ”seen enough” and is sufficiently sophisticated (see fig. 23).

## VI. Challenges and Further Work

One main limitation on model training and active learning was the lack of support for parallelism, which significantly limited the scope and efficiency of experiments. Both DataParallel and DataDistributed methods (enabled by Pytorch APIs) were experimentally implemented in the codebase, but the main issue was with the model performance, where the model does not seem to learn and update its weights with each minibatch, and tweaking with the hyperparameters(such as learning rate) did not solve the problem.

Another drawback was that the full experiment setup did not use the OSCAR base model trained with its own predefined loss function. This is mostly because we altered the model structure, yet pretraining on our own was too resource-consuming to perform. Furthermore, due to the memory constraints and the size of multimodal data entries, very small batch sizes had to be used to avoid memory crash during training.

Two main extra feature groups - sentiment features and correlation features - were explored in the experiments. Several other features were also considered but not included in our experiments for various reasons. OCR results, for one, were very noisy given our setup, such that they were excluded for most of our explorations. Knowledge Graph Entity was another useful tool for locating key ’tags’ in the images powered by google cloud (see here). Nevertheless, the recognition inference time is prohibitively long, and the current API is more suitable for single-entry inference than batch processing.

In the active learning setup, there are also a few directions that can be further pursued with future work. Firstly, although we adopted both MC dropout and ensemble methods for uncertainty estimation with their results presented, no direct comparisons between the two groups of approaches were produced. Secondly, Conducting more experiments with the full dataset and a smaller step size (much smaller than 5000) could more clearly illustrate the point at which the model has a comparable performance as the model trained on the full dataset as well as showing a finer active learning curve. Finally, a stopping criteria can be further developed to save the effort of labelling unnecessarily large dataset. Similar to early stopping in model training, a stopping scheme can be devised for active learning [15] when the model performances and/or uncertainty estimates have stagnated or plateaued.

## VII. Conclusion

This report shows the promising results of the early experiments with transformer-based multi-modal methods for hate speech detection. Based on the previous work of anchoring the text and image inputs with object tags as the alignment, we further proposed our own task-specific features with extra model layers to incorporate both the multi-modal model design and the extracted higher-lever representations of the speech characteristics tailored for hate classification. The results, slightly better than SOTA built upon LSTM algorithms, however, are still behind the human accuracy of over 80%. Also, although the proposed multimodal model outperformed the Text-only baselines, we did not see a boost in other metrics, such as ROC-AUC.

Furthermore, an active learning procedure was implemented for this multi-modal task to study the feasibility of reducing the efforts of labelling for a large dataset. With a variety of uncertainty estimate methods, we can show that comparable or better performances with less than 30% of the full dataset labelled could be enabled with carefully chosen active learning setups.

## References

[1] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," 2019.

[2] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," 2020.

[3] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," 2019.

[4] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," 2019.

[5] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," 2020.

[6] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, "The hateful memes challenge: Detecting hate speech in multimodal memes," 2021.

[7] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, Y. Choi, and J. Gao, "Oscar: Object-semantics aligned pre-training for vision-language tasks," 2020.

[8] R. Gomez, J. Gibert, L. Gomez, and D. Karatzas, "Exploring hate speech detection in multimodal publications," 2019.

[9] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," 2021.

[10] P. Atighehchian, F. Branchaud-Charron, J. Freyberg, R. Pardinas, and L. Schell, "Baal, a bayesian active learning library," https://github.com/ElementAI/baal/, 2019.

[11] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," 2016.

[12] A. Kirsch, J. van Amersfoort, and Y. Gal, "Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning," 2019.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[14] D. Beck, L. Specia, and T. Cohn, "Reducing annotation effort for quality estimation via active learning," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 543–548. [Online]. Available: https://aclanthology.org/P13-2097

[15] Z. Pullar-Strecker, K. Dost, E. Frank, and J. Wicker, "Hitting the target: Stopping active learning at the cost-based optimum," 2021.

## Appendix A
## More experiment results

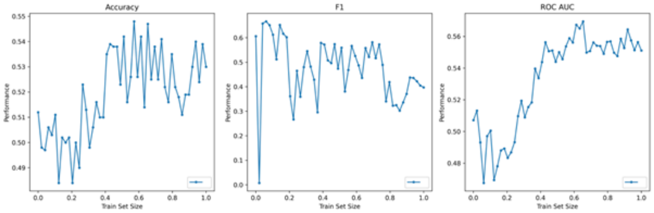


Fig. 13. Multi-modal active learning step-wise performance using the proposed model with a setup of 5000 samples in total and a step size of 100. The uncertainty estimation method is random selection(baseline) with a cumulative training sample pool.

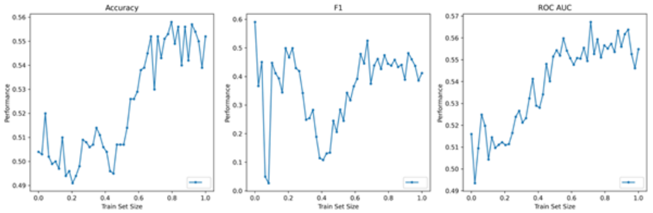


Fig. 14. Multi-modal active learning step-wise performance using the proposed model with a setup of 5000 samples in total and a step size of 100. The uncertainty estimation method is maximized entropy (based on MC-dropout) with a cumulative training sample pool.

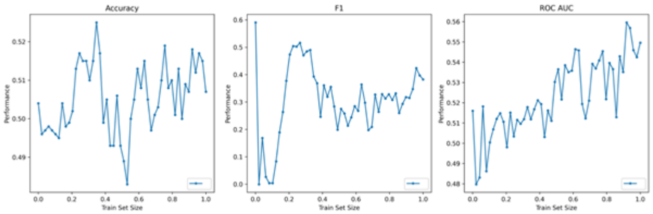


Fig. 15. Multi-modal active learning step-wise performance using the proposed model with a setup of 5000 samples in total and a step size of 100. The uncertainty estimation method is maximized BALD (based on MC-dropout) with a cumulative training sample pool.

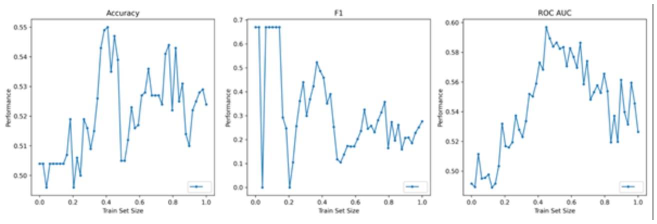


Fig. 16. Multi-modal active learning step-wise performance using the proposed model with a setup of 5000 samples in total and a step size of 100. The uncertainty estimation method is maximized variation ratio (based on ensemble method) with a cumulative training sample pool.

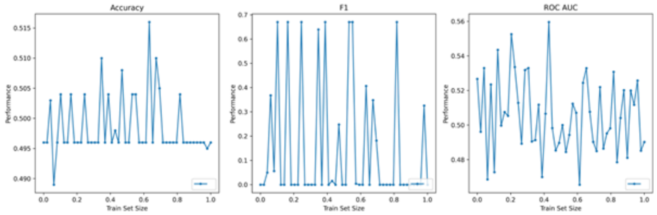


Fig. 17. Multi-modal active learning step-wise performance using the proposed model with a setup of 5000 samples in total and a step size of 100. The uncertainty estimation method is maximized mutual information (based on ensemble method) with checkpoints (training pool of the freshly selected samples only).

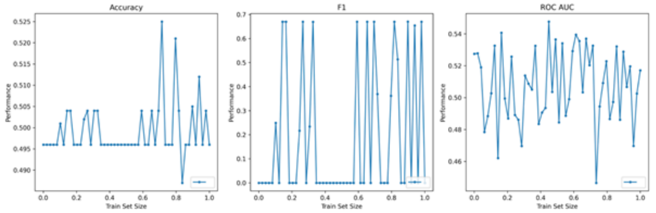


Fig. 18. Multi-modal active learning step-wise performance using the proposed model with a setup of 5000 samples in total and a step size of 100. The uncertainty estimation method is maximized variation ratio (based on ensemble method) with checkpoints (training pool of the freshly selected samples only).

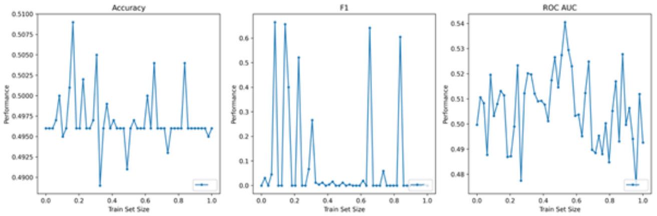


Fig. 19. Multi-modal active learning step-wise performance using the proposed model with a setup of 5000 samples in total and a step size of 100. The uncertainty estimation method is random selection (baseline) with checkpoints (training pool of the freshly selected samples only).

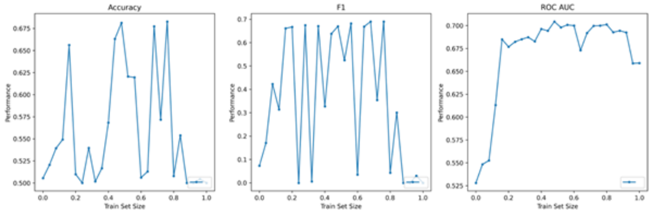


Fig. 20. Multi-modal active learning step-wise performance using the proposed model with a setup of the full dataset as the unlabelled pool (120000+ samples) and a step size of 5000. The uncertainty estimation method is maximized entropy (based on MC-dropout) with a cumulative training sample pool.

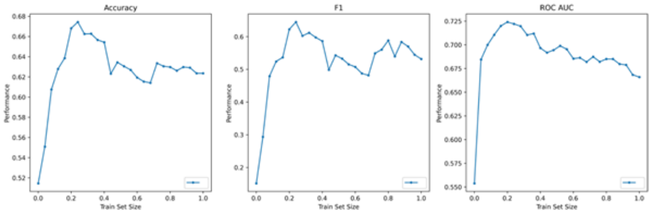


Fig. 21. Multi-modal active learning step-wise performance using the proposed model with a setup of the full dataset as the unlabelled pool (120000+ samples) and a step size of 5000. The uncertainty estimation method is random selection (baseline) with a cumulative training sample pool.

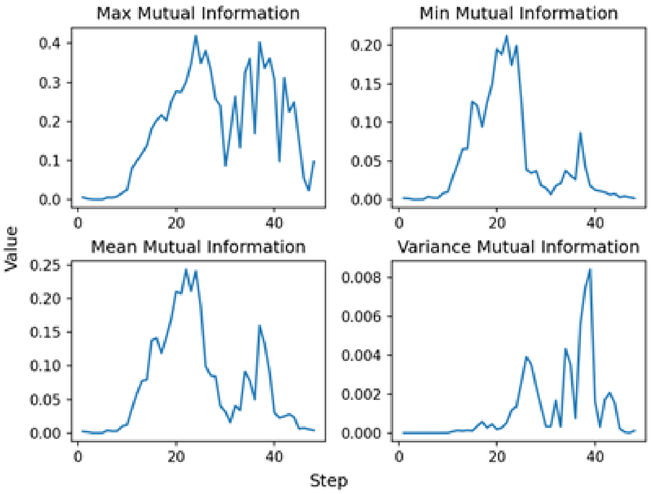


Fig. 22. Step-wise statistics of mutual information evaluated on the remaining unlabelled pool.

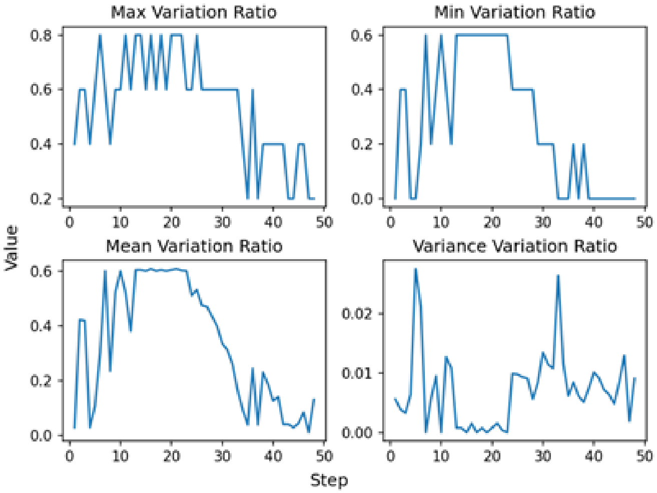


Fig. 23. Step-wise statistics of variation ratio evaluated on the remaining unlabelled pool.

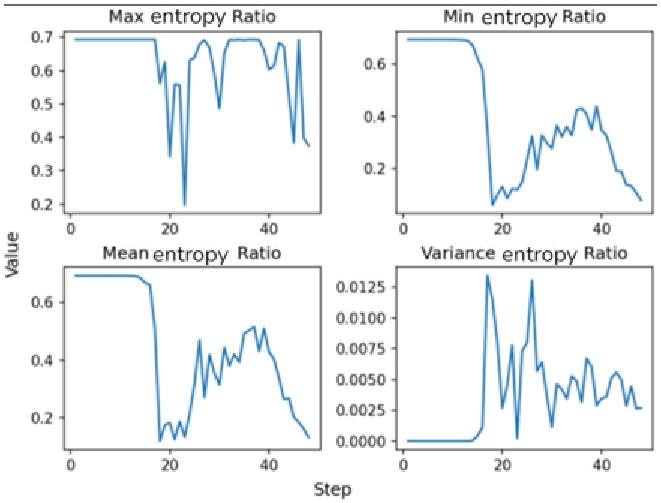


Fig. 24. Step-wise statistics of expected entropy evaluated on the remaining unlabelled pool.