Taming the Opinionated Transformer: Detecting Out-of-Set Data Using Uncertain Labels

Anonymous ACL submission

Abstract

Even models that demonstrate strong performance during development may crash and burn in production, resulting in end-user mistrust and potential adverse outcomes. Many such failures "in the wild" are due to the often unexpected yet common occurrence of data mismatch between training and inference. We present a method to train a transformer model to return ambiguous predictions in the face of unfamiliar data. The performance of this model on a multi-label classification task is on-par with traditional approaches. Additionally, it can better identify and reject out-of-set examples supplied at inference time, making it a promising approach for real-world deployment.

1 Introduction

Transformer models are experiencing increasing adoption in industry settings due to state-of-the-art performance for multiple NLP applications (Devlin et al., 2019; Winata et al., 2021). As these sophisticated models are released “in the wild,” even models that performed well during development often produce sub-optimal predictions (Talby, 2019). In a best-case scenario, the model has been deployed as part of a human-in-the-loop application, and the end-users will learn to mistrust certain predictions even if it means a decrease in productivity. In a worst-case scenario, an automatic decision making process will use the poor predictions to generate outcomes that cause adverse economic and societal downstream effects.

Many failures of models “in the wild” can be attributed to the simple fact that the data supplied during inference did not closely resemble the training data. This situation may arise for a variety of reasons: The data pipeline used in production may not match that used to obtain data for training, or upstream changes to the data pipeline may suddenly occur (Sculley et al., 2015). Alternatively, the end-user(s) of the system may mistakenly input erroneous data. One phenomenon that has received much attention is that of model drift, wherein data evolves and diverges from what was present at the time a model was trained (Dube and Farchi, 2020; Gama et al., 2014; Barros and Santos, 2018). One way to deal with these scenarios is to develop a model that knows when it does not know. In other words, a model that is able to indicate uncertainty when data is presented that is different from the data presented during training.

In this paper, we present a method for training transformer models for multi-label text classification that not only performs well for in-set examples (i.e., those similar to the training data distribution), but also demonstrates an appropriate agnosticism when making predictions for out-of-set examples that do not resemble the training data. A model trained in this way is better able to identify unusual or unexpected examples supplied at inference time while not adversely affecting predictions on expected data. Furthermore, detection of drift is intrinsic to the model itself, facilitating a faster path to production monitoring by not requiring a separate data monitoring solution. This approach has clear benefits for production-grade models and active learning, making it a promising technique for NLP tasks deployed "in the wild."

2 Related Work

The state of model and data monitoring in production is premature and rapidly evolving. Proposed solutions may require data-specific metrics to be developed and maintained for each application (Breck et al., 2019), while others focus on visualization and exploratory data analysis (Baniecki et al., 2021). Alternatively, metrics based on trained model outputs may be developed to detect a change in distribution (Dube and Farchi, 2020).

One issue that makes out-of-set detection difficult stems from the tendency for deep learning models to be "opinionated," often returning strong

positive or negative predictions even for unfamiliar data. In the computer vision domain, this challenge has been discussed in the context of "open set" image classification (Bendale and Boult, 2015). Dhamija et al. (2018) propose a framework for model training that computes loss separately for "foreground" and "background" images, encouraging agnostic labels for the background case. We take this work as the inspiration for our approach.

3 Experimental Setup

3.1 Model Development

We train a five-class multi-label classifier to assign text examples to one or multiple categories. We use a RoBERTa (Liu et al., 2019) foundation model available through huggingface.co[1] as the base model, with a sequence classification linear layer on top. We train the model on our data for one additional epoch using a batch size of 8.

To develop our model, we follow an approach inspired by Dhamija et al. (2018), but modified for multi-label text classification with transformers rather than multi-class image classification using CNNs. Specifically, we construct a training data set that includes both examples from five classes of interest ("In-Set Positive") and examples that do not belong to any of the five classes ("In-Set Negative"). For examples belonging to the In-Set Positive set, loss is computed using a standard binary cross-entropy loss term. For In-Set Negative examples, we compute the mean-squared error of the logits in the penultimate layer of the network:

$$L(x) = 1/N \sum_{n=1}^{N} |x|^2 \quad (1)$$

When a sigmoid activation is applied on top of this layer during inference, the effect is to drive all class outputs for background examples to 0.5. This model is referred to throughout the rest of the paper as the ambiguous-aware (AA) model.

For comparison, we also train two models using standard approaches:

- Positive (P): A five-class multi-label classifier trained using only in-set positive examples.
- Positive + Negative (P+N): A five-class multi-label classifier trained using examples both from the in-set positive and negative sets, but with no change to the loss function (i.e., binary cross-entropy loss computed for all examples).

3.2 Data Sets

For model training and evaluation, we use the Apte-Mod version of the Reuters-21578 corpus[2], available for download through nltk[3] (v. 3.6.5). This data set contains text snippets from the Reuters newswire in 1987, along with one or more text categories assigned to each snippet. We divide examples into three partitions according to labels:

- In-Set Positive - Examples containing at least one of the top five most prevalent labels
- In-Set Negative - Examples that are not included in the In-Set Positive set, and that contain at least one of the 6th-10th most prevalent labels
- Reuters-OOS (Out-of-Set) - Remaining examples that are neither in the In-Set Positive nor the In-Set Negative set

Examples from In-Set Positive and In-Set Negatvie are both used during training, but Reuters-OOS examples are withheld until inference time. The purpose of Reuters-OOS is to test whether the trained models can distinguish examples that are similar to the training examples but nonetheless represent some divergence in language.

For inference, we also use the Sentiment Polarity data set Version 2.0 (Pang and Lee, 2004), which contains movie reviews and is available through nltk. These data are used to test whether the trained models can distinguish language that contains surface-level similarity to the Reuters examples (i.e., short paragraphs of English language text), but bears little to no resemblance to the tone and vocabulary of the training data. We refer to this data set throughout as Movies-OOS.

Table 1 tabulates the number of examples belonging to each of the described data sets. For data used during training, the split between train and test examples is also shown.

3.3 Evaluation

To evaluate each model, we consider two criteria:

[1] https://huggingface.co/transformers/pretrained_models.html

[2] The Reuters-21578, Distribution 1.0 test collection is available from David D. Lewis' professional home page, currently: http://www.research.att.com/~lewis

[3] https://www.nltk.org/

	Train	Test
In-Set Positive	5,837	2,310
In-Set Negative	652	235
Reuters-OOS	-	1,754
Movies-OOS	-	2,000

Table 1: Number of examples belonging to each of the data sets.

	Model		
	P	**P+N**	**AA**
Precision	0.983	0.989	0.984
Recall	0.975	0.958	0.968
Background accuracy	0.209	0.796	0.574

Table 2: Performance metrics for each of the three trained models.

1. How does the model perform on in-set data?

2. Do the model predictions allow us to discriminate between in-set and out-of-set data?

3.3.1 Model Performance on In-Set Data

For all models, we compute the precision, *P*, and recall, *R*, using the In-Set Positive test set, microaveraged across all *M* classes:

$$P = \frac{\sum_{m=1}^{M} TP_m}{\sum_{m=1}^{M} (TP_m + FP_m)} \quad (2)$$

$$R = \frac{\sum_{m=1}^{M} TP_m}{\sum_{m=1}^{M} (TP_m + FN_m)} \quad (3)$$

Additionally, we report the "background accuracy," defined as the fraction of examples from the In-Set Negative test set that are correctly classified (i.e., none of the five labels are predicted).

3.3.2 Detecting Out-of-Set Predictions

We compute the area under the receiver operating curve (AUC) to measure the discriminative ability of each model. In this case, AUC measures the model's ability to discriminate between in-set and out-of-set examples, rather than its traditional use in differentiating positive and negative examples. An AUC of 1.0 indicates that the we can perfectly distinguish between the in- and out-of-set data distributions, while 0.5 indicates a complete inability to differentiate between the two.

AUC is computed both on a per-class and per-example basis. To compute per-class AUC, each of the five class membership scores output from the final sigmoid layer for each example are labeled as in-set or out-of-set based on the example the score originated from. For in-set examples, scores would ideally lie close to 0 or 1, indicating high confidence of class membership or non-membership, whereas for out-of-set examples, scores should lie closer to 0.5, indicating ambiguity in the face of unfamiliar data. Therefore, AUC is computed on the absolute distance of each score from 0.5 rather than the raw score itself.

To compute per-example AUC, the squared Euclidean norm of the five absolute distance scores for each example is used. This reflects the Euclidean distance from each example's class membership score vector to a vector containing only "ambiguous" labels, i.e., a vector of class membership scores of 0.5 for all classes.

4 Results

Table 2 displays model performance metrics for each of the three trained models. Precision and recall maintain similarly high values across all three models, indicating no large degradation in performance due to differences in the models. For both the P+N and AA models, the background accuracy metric shows that the majority of background examples are correctly assigned no labels. Background accuracy is substantially worse for model P, which incorrectly assigns at least one label to the majority (79%) of background examples.

Figure 1 displays histograms of output scores for In-Set Positive vs. Reuters-OOS (top) and Movies-OOS (bottom) data across the three models. Both the P and P+N models score out-of-set data similarly to negative class membership scores for in-set data, assigning both a score close to 0.0. The AA model, however, tends to return high-confidence class membership scores (i.e., scores that are close to 0.0 or 1.0) for in-set data only, and scores near 0.5 for out-of-set data. Note that the AA model was not provided any data from the out-of-set category during training but nonetheless returns "uncertain" scores for these examples.

Table 3 displays the AUC values obtained from comparison of in-set vs. out-of-set data, both using raw class membership scores and scores aggregated at a per-example level. Across score types and data sets, the P+N model consistently yields the lowest AUC values. The P model performs well on Movies-OOS, but AUC drops using Reuters-OOS.

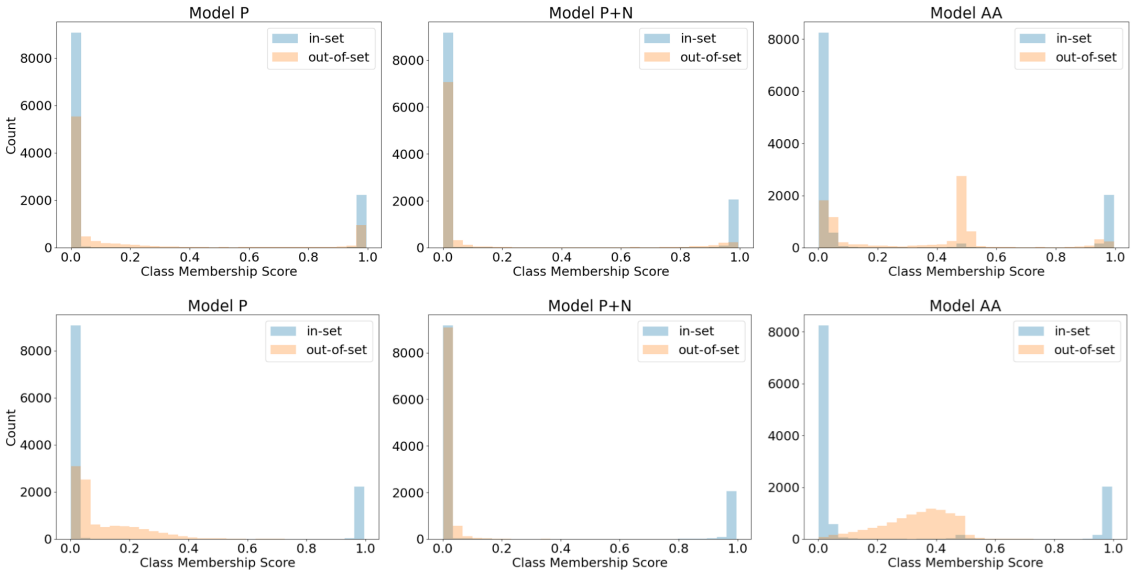

Figure 1: Histograms of output scores for in-set vs. Reuters-OOS (top) and Movies-OOS (bottom) data across the three models.

	Per-Class			Per-Example		
	P	**P+N**	**AA**	**P**	**P+N**	**AA**
Reuters-OOS	0.843	0.739	0.928	0.919	0.894	0.944
Movies-OOS	0.974	0.788	0.972	0.988	0.825	0.976

Table 3: AUCs from comparison of in-set vs. out-of-set data.

The AA model maintains consistently high AUC across the two data sets and score types.

5 Discussion

We have demonstrated a method to train an "ambiguous-aware" transformer for multi-label classification that intrinsically supports robustness to unfamiliar data. Compared with traditional approaches, the AA model can better distinguish out-of-set data across the two data sets. Notably, this model outperforms the second-best approach (P) on the Reuters-OOS data set, which is more similar to the training samples. The subtle differences between in-set and Reuters-OOS resemble differences arising from model drift. Movies-OOS is more dissimilar to the in-set data, mimicking data deviations that may occur due to erroneous user inputs or changes to the data pipeline.

Past work has explored using raw confidence values to identify out-of-set data (Hendrycks and Gimpel, 2016), an approach that we test using Model P. One practical consideration is selecting an appropriate confidence threshold for flagging data as likely out-of-set. As shown in Figure 1, a clear distinction between the in-set and out-of-set distributions exists for Model AA only, with Model P demonstrating no clear boundary between the two sets.

One advantage of the AA training approach is that detection of unfamiliar data is intrinsic to the trained model itself rather than a separate solution. Thus, out-of-set data can be identified and monitored without necessitating additional custom, data-specific metrics. While custom metrics may remain important and useful, our experience is that they are often developed after-the-fact and may not be released with the first version of a model. This method of data monitoring, however, can be introduced as part of an initial deployment with little additional development or maintenance overhead.

6 Conclusion

We have trained a transformer for multi-label text classification that not only performs well on the task at hand, but that can also distinguish out-of-set data better than traditional approaches. This training approach yields production-ready models that are reliable and trustworthy, even in the face of unfamiliar data.

References

Hubert Baniecki, Wojciech Kretowicz, Piotr PiÄ…tyszek, Jakub WiÅ›niewski, and PrzemysÅ‚aw Biecek. 2021. dalex: Responsible machine learning with interactive explainability and fairness in python. *Journal of Machine Learning Research*, 22(214):1–7.

Roberto Souto Maior Barros and Silas Garrido T. Carvalho Santos. 2018. A large-scale comparison of concept drift detectors. *Information Sciences*, 451-452:348–370.

Abhijit Bendale and Terrance E. Boult. 2015. Towards open set deep networks. *CoRR*, abs/1511.06233.

Eric Breck, Marty Zinkevich, Neoklis Polyzotis, Steven Whang, and Sudip Roy. 2019. Data validation for machine learning. In *Proceedings of SysML*.

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.

Akshay Raj Dhamija, Manuel Günther, and Terrance E. Boult. 2018. Reducing network agnostophobia. *CoRR*, abs/1811.04110.

Parijat Dube and Eitan Farchi. 2020. *Automated Detection of Drift in Deep Learning Based Classifiers Performance Using Network Embeddings*, pages 97–109.

João Gama, Indrundefined Žliobaitundefined, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4).

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *CoRR*, abs/1610.02136.

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.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

David Talby. 2019. Why machine learning models crash and burn in production. *Forbes*.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.