Machine Learning (2022) 111:4539–4563
https://doi.org/10.1007/s10994-022-06263-z

Adversarial examples for extreme multilabel text classification

Mohammadreza Qaraei[1] · **Rohit Babbar**[1]

Received: 15 December 2021 / Revised: 19 July 2022 / Accepted: 29 September 2022 /
Published online: 4 November 2022

Abstract
Extreme Multilabel Text Classification (XMTC) is a text classification problem in which, (i) the output space is extremely large, (ii) each data point may have multiple positive labels, and (iii) the data follows a strongly imbalanced distribution. With applications in recommendation systems and automatic tagging of web-scale documents, the research on XMTC has been focused on improving prediction accuracy and dealing with imbalanced data. However, the robustness of deep learning based XMTC models against adversarial examples has been largely underexplored. In this paper, we investigate the behaviour of XMTC models under adversarial attacks. To this end, first, we define adversarial attacks in multilabel text classification problems. We categorize attacking multilabel text classifiers as (a) positive-to-negative, where the target positive label should fall out of top-k predicted labels, and (b) negative-to-positive, where the target negative label should be among the top-k predicted labels. Then, by experiments on APLC-XLNet and AttentionXML, we show that XMTC models are highly vulnerable to positive-to-negative attacks but more robust to negative-to-positive ones. Furthermore, our experiments show that the success rate of positive-to-negative adversarial attacks has an imbalanced distribution. More precisely, tail classes are highly vulnerable to adversarial attacks for which an attacker can generate adversarial samples with high similarity to the actual data-points. To overcome this problem, we explore the effect of rebalanced loss functions in XMTC where not only do they increase accuracy on tail classes, but they also improve the robustness of these classes against adversarial attacks. The code for our experiments is available at https://github.com/xmc-aalto/adv-xmtc.

Keywords Extreme classification · Adversarial attacks · Multilabel problems · Text classification · Data imbalance

Editors: Emilie Devijver and Krzysztof Dembczynski.

✉ Rohit Babbar
rohit.babbar@aalto.fi

[1] CS Department, Aalto University, Helsinki, Finland

1 Introduction

Extreme Multilabel Text Classification (XMTC) addresses the problem of tagging text documents with a few labels from a large label space, which has a wide application in recommendation systems and automatic labelling of web-scale documents (Partalas, Kosmopoulos, Baskiotis, Artieres, Paliouras, Gaussier, Androutsopoulos, Amini, and Galinari, 2015; Jain, Balasubramanian, Chunduri, and Varma, 2019; Agrawal, Gupta, Prabhu, and Varma, 2013). There are three characteristics which make XMTC different from typical text classification problems: XMTC is a multilabel problem, the output space is extremely large, and data are highly imbalanced following a power-law distribution (Babbar, Metzig, Partalas, Gaussier, and Amini, 2014), which makes models perform poorly on a large fraction of labels with few training samples, known as tail labels (see Fig. 1).

The research on XMTC has focused on tackling the aforementioned challenges by proposing models which can scale to millions of labels (Babbar & Schölkopf, 2017; Jain, Balasubramanian, Chunduri, and Varma, 2019; Prabhu, Kag, Harsola, Agrawal, and Varma, 2018; Medini, Huang, Wang, Mohan, and Shrivastava, 2019) and mitigating the power-law impact on predicting tail classes by rebalancing the loss functions (Qaraei, Schultheis, Gupta, and Babbar, 2021; Cui, Jia, Lin, Song, and Belongie, 2019). However, as XMTC algorithms have shifted from shallow models on bag-of-words features to deep learning models on word embeddings (You, Zhang, Wang, Dai, Mamitsuka, and Zhu, 2019; Ye, Chen, Wang, and Davison, 2020; Jiang, Wang, Sun, Yang, Zhao, and Zhuang, 2021), two new questions need to be addressed : (i) how can one perform adversarial attacks on XMTC models, and (ii) how robust are these models against the generated adversarial examples? These questions are also the key to understanding the explainability of modern deep learning models.

Adversarial attacks are performed by applying engineered noise to a sample, which is imperceptible to humans but can lead deep learning models to misclassify that sample. While the robustness of deep models to adversarial examples for image classification problems has been extensively studied (Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, and Fergus, 2014; Goodfellow, Shlens, and Szegedy, 2015), corresponding methods for generating adversarial examples have also been developed for text classification by taking into account the discrete nature of language data (Zhang, Sheng,

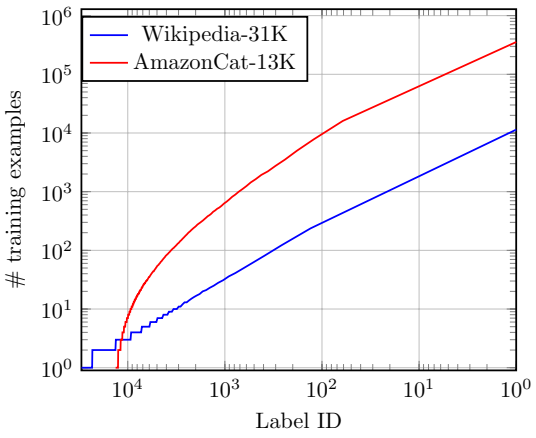


Fig. 1 Label frequency of two XMTC datasets, Wikipedia-31K and AmazonCat-13K. Both datasets have an extremely imbalanced distribution, where the frequencies of a few head labels are high, but there are only a few training samples for a large fraction of labels known as tail classes

Table 1 An adversarial example generated for APLC-XLNet by targeting the tail label “shorthand" of the Wikipedia-31K dataset. While “shorthand" is among top-5 predicted labels for the real sample, it will become the 7th predicted label by only replacing “executives” with “companies". Notably, the newly predicted label “history" is not one of the true labels

Sample	Gregg shorthand is a form of shorthand that was invented by John Robert Gregg in 1888 ... With the invention of dictation machines, shorthand machines, and the practice of companies writing their own letters on their personal computers .
True labels	article, collage, cool, education, graphics, gregg, language, orthography, reference, shorthand, speed, stenography, tools, wiki, wikipedia, writing
Pred. (Real)	language, writing, wikipedia, typography, **shorthand**, history, wiki, linguistics
Pred. (Adv.)	language, writing, wikipedia, typography, history, wiki, **shorthand**, english

Alhazmi, and Li, 2020). However, the research on adversarial attacks on text classifiers is limited to small to medium scale datasets, and the tasks are binary or multiclass problems, making current adversarial frameworks not applicable in XMTC.

In this paper, we explore adversarial attacks on XMTC models. To this end, inspired by Song et al. (2018) and Hu et al. (2021), first we define adversarial attacks on multilabel text classification problems. Two types of attacks that can happen in the real world on multilabel text classification models is to: (i) manipulate a sample to drop a positive label from the top-k predicted labels, called positive-to-negative and (ii) make the model predict a negative label as a positive label by pushing the targeted label into the top-k predicted labels, called negative-to-positive attacks, in this paper. For instance, in a recommendation system, a malicious company may try to prevent the products of their rival companies from being recommended by the model by manipulating the description of their product, which can be categorized as a positive-to-negative attack. Also, in the negative-to-positive case, a malicious company may manipulate the description of their product in order to fool the model to have their own products among the recommended ones as much as possible. After introducing the attacks, in a setting limited to top-5, we show that XMTC models, in particular the attention-based AttentionXML (You, Zhang, Wang, Dai, Mamitsuka, and Zhu, 2019) and the transformer-based APLC-XLNet (Ye, Chen, Wang, and Davison, 2020), are vulnerable to positive-to-negative adversarial attacks, but more robust to negative-to-positive attacks.

Our analysis also shows that the success rate of the adversarial attacks on XMTC models has imbalanced behaviour, similar to the distribution of the data. In particular, our experiments show that positive tail classes are very easy to attack. This means that not only is it difficult to correctly predict a tail label, but also there is a high chance that one can eliminate a correctly classified tail label from the predicted labels by changing a few words, or even a single word in some cases (Table 1).

To improve robustness of tail classes against adversarial attacks, we investigate the rebalanced loss functions originally proposed to enhance model performance on missing/infrequent labels (Qaraei, Schultheis, Gupta, and Babbar, 2021). Our results show these loss functions can significantly increase robustness of correctly (when trained with vanilla loss) classified samples belonging to tail classes. We show that part of the increase in the robustness of tail classes comes from the higher scores of these labels when a rebalanced loss is used compared to a normal loss.

We also measure unnoticeability of positive-to-negative attacks from two points of views: the similarity of the generated adversarial samples to the real samples in terms of their word embeddings and change rates needed to generate the adversarial samples, and also by measuring the changes in the predicted labels, excluding the targeted label, after the attacks. To measure the latter, we compute the overlap between the top-5 predicted labels for a real sample and its corresponding adversarial sample and also the changes in the rank of the predicted labels before and after the attack using the normalized discounted cumulative gain (nDCG) metric. Both of the approaches show high unnoticeability of the positive-to-negative attack.

To summarize, the key findings of our work are:

- XMTC models are vulnerable to positive-to-negative adversarial examples, as shown by experiments with AttentionXML and APLC-XLNet.
- The generated adversarial samples in positive-to-negative attacks have high similarity to the real samples, and the attacks are highly unnoticeable.
- The success rate of the adversarial attacks on XMTC has an imbalanced behaviour similar to the distribution of data, where it is easy to attack positive tail labels by only changing a few words in the corresponding samples.
- The rebalanced loss functions can significantly improve the robustness of tail labels against positive-to-negative adversarial attacks.

2 Related work

2.1 Adversarial attacks on text classifiers

Adversarial attacks on image classification cannot be directly applied to text classification problems because of the discrete structure of text data. In text classification problems, this is achieved by first finding important parts of the text and then manipulating these parts. In white-box attacks, the important parts are determined by the gradient information, and in black-box attacks this is done by masking some parts of the text and then computing the difference between the output probabilities of the masked and unmasked sample. A perturbation that is undetectable for humans should result in a sample that is semantically similar to the original and preserve the fluency of the text.

Adversarial attacks in text classification problems can be categorized into character-level and word-level attacks. In Sun et al. (2020) and Li et al. (2018), which are two character-level attacks, first, important words are determined by the gradient magnitude and then those words are misspelled to change the label. The same idea is used in Gao et al. (2018) but with a black-box setting for finding the important words. The problem with the character-level methods is that a spell checker can easily reveal the adversarial samples.

In word-level attacks, Jin et al. (2020) and Ren et al. (2019) find the important words in a black-box setting and then replace those words with synonyms to change the predicted label. However, the substituted words may damage the fluency of the sentences. To preserve the fluency of the sentences, a language model such as BERT (Devlin, Chang, Lee, and Toutanova, 2018) can be used to generate the candidates in a context-aware setting (Garg & Ramakrishnan, 2020; Li, Ma, Guo, Xue, and Qiu, 2020; Xu & Veeramachaneni, 2021).

Current works in adversarial attacks on text classifiers have focused on binary or multiclass problems without taking into account data irregularities such as imbalanced data distribution for instance. For attacking XMTC models, first we extend the adversarial framework for finding important words to multilabel settings, then we use the BERT model (Li, Ma, Guo, Xue, and Qiu, 2020) for context-aware word substitutions which can generate fluent adversarial samples.

2.2 XMTC models

Earlier works in XMTC used shallow models on bag-of-words features (Bhatia, Jain, Kar, Varma, and Jain, 2015; Babbar & Schölkopf, 2017; Khandagale, Xiao, and Babbar, 2020). However, as bag-of-words representation looses contextual information, recent XMTC models employ deep neural networks on word embbedings. Among these models, AttentionXML (You, Zhang, Wang, Dai, Mamitsuka, and Zhu, 2019) uses a BiLSTM layer followed by an attention module over pretrained word embbedings and is trained in a tree-structure to reduce the computational complexity. APLC-XLNet (Ye, Chen, Wang, and Davison, 2020) is a transformer-based approach, which fine tunes XLNet (Yang, Dai, Yang, Carbonell, Salakhutdinov, and Le, 2019) on extreme classification datasets. To scale XLNet to a large number of labels, APLC-XLNet partitions labels based on their frequencies, and the loss for most of the samples is computed only on a fraction of these partitions.

Another major challenge in XMTC is the problem of infrequent and missing labels (Jain, Prabhu, and Varma, 2016; Qaraei, Schultheis, Gupta, and Babbar, 2021). To improve generalization on infrequent labels, ProXML (Babbar & Schölkopf, 2019) optimizes squared hinge loss with ℓ_1 regularization. To address the missing labels problem, Jain et al. (2016) optimizes a propensity scored variant of normalized discounted cumulative gain (nDCG) in a tree classifier. Qaraei et al. (2021) propose to reweight popular loss functions, such as BCE and squared hinge loss to make them convex surrogates for the unbiased 0-1 loss. The reweighted loss functions are further rebalanced by a function of label frequencies to improve performance on tail classes. Our experiments show that, even though these losses were not designed from a robustness perspective but more from the viewpoint of being statistically unbiased under missing labels, they significantly improve the robustness of tail classes against adversarial attacks in deep XMTC models.

2.3 Adversarial attacks on imbalanced or multilabel problems

Adversarial attacks on multilabel problems were first defined in Song et al. (2018) for multilabel classification or ranking. Hu et al. (2021) used a different approach for attacking multilabel models by proposing loss functions which are based on the top-k predictions. In Melacci et al. (2020), the domain knowledge on the relationships among different classes are used to evade adversarial attacks against multilabel problems. Yang et al. (2020) defined the attackability of mulitlabel classifiers, and proved that the spectral norm of a classifier's parameters and its performance on unperturbed data are two key factors in this regard.

Adversarial attacks on models trained on imbalanced data were discussed in Wang et al. (2021) and Wu et al. (2021). In both works, it has been remarked that for adversarially trained models, the decay of accuracy from head to tail classes on both clean and adversarial examples are more than that of normal training. To overcome this problem, Wu et al. (2021) used a margin-based scale-invariant loss to deal with imbalanced

distribution, along with a loss to control the robustness of the model. Wang et al. (2021) showed that rebalancing robust training can increase the accuracy of tail classes but has significant adverse effects on head classes. To tackle this problem, they proposed to use reweighted adversarial training along with a loss which makes features more separable.

All the aforementioned works are on image classification problems. To the best of our knowledge, adversarial attacks on multilabel problems in the text classification domain have only been explored in Wu et al. (2017) and Song et al. (2021), where the former did not aim to produce adversarial examples, but to use adversarial training to improve accuracy, and the latter was an introduction on the relationship between multilabel classification and adversarial attacks.

3 Adversarial attacks on multilabel text classifiers

Attacking multilabel problems is different from binary or multiclass problems since the samples in the first may have multiple positive labels, and therefore a manipulated sample can be an adversarial sample for some labels but not for the others. While attacking multilabel image classification models has been recently explored in Song et al. (2018) and Hu et al. (2021), to the best of our knowledge, there is no work on attacking multilabel problems in the NLP domain.

In this section, first, we define adversarial attacks on multilabel text classifiers. This definition is close to that of Song et al. (2018) in the computer vision domain, in which there are non-targeted and targeted attacks, where in non-targeted attacks the goal is to change at least one (non-specified) label, while in targeted attacks one tries to either make a specific positive label as negative or vice versa. After defining multilabel adversarial attacks, we discuss how to perform the attacks on text classifiers. To this end, we extend calculating word importance in Jin et al. (2020) to the multilabel case to be consistent with our definition of multilabel attacks, then the same procedure as Jin et al. (2020) for word substitution using the Bert model is employed until the goal of the attack is reached.

Assume $S = [w_1, ..., w_n]$ is a document consisting of n words $w_i \in \mathbb{R}^d$, $\mathrm{y} \in \{0, 1\}^L$ are the labels corresponding to this document in one-hot encoded format, and $Y = \{i | y_i = 1\}$ represents indices of the positive labels. Let $g : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^L$ be a mapping from documents to scores, where $g_i(S) \in R$ indicates the score of the i-th label. Also, $\hat{Y}_k(S) = \{T_1(g(s)), ..., T_k(g(s))\}$ represents the top-k predicted labels, where $T_i : \mathbb{R}^L \rightarrow \{1, ..., L\}$ is an operator which returns the index of the i-th largest value. The goal in adversarial attacks is to generate a document S' which is similar to S but has different predicted labels.

As in multiclass classification, we can also have non-targeted and targeted attacks on multilabel problems, which are defined in the following.

Table 2 Summary of targeted attacks

Attack type	A_1	A_{-1}	Goal
Pos-to-neg	$\{i : i \in Y, i \in \hat{Y}_k(S)\}$	$\emptyset$	$a_1 \notin \hat{Y}_k(S'), a_1 \sim A_1$
Neg-to-pos	$\emptyset$	$\{i : i \notin Y, i \notin \hat{Y}_k(S)\}$	$a_{-1} \in \hat{Y}_k(S'), a_{-1} \sim A_{-1}$

3.1 Non-targeted attacks

In a non-targeted attack, the goal is to replace at least one label among top-k predictions with a label from out of this set. This can be formulated as the following optimization problem:

$$\begin{aligned} \arg\min_{S'} \quad & d(S, S') \\ \text{s.t.} \quad & \exists i \in \hat{Y}_k(S') : i \notin \hat{Y}_k(S) \end{aligned} \tag{1}$$

where $d(.,.)$ is a distance metric which can be interpreted as the inverse of the similarity.[1]

3.2 Targeted attacks

In a targeted attack on a multilabel problem, an attacker may try to decrease the score of a particular positive label (positive-to-negative attacks), or increase the score of a negative label in order to be among the top-k predicted labels (positive-to-negative attacks). These can be formulated as follows:

$$\begin{aligned} \arg\min_{S'} \quad & d(S, S') \\ \text{s.t.} \quad & a_1 \notin \hat{Y}_k(S'), \quad a_1 \sim A_1 \\ & a_{-1} \in \hat{Y}_k(S'), \quad a_{-1} \sim A_{-1} \end{aligned} \tag{2}$$

where a_1 and a_{-1} are the targeted labels selected (denoted by $\sim$) from the sets A_1 and A_{-1}, respectively. For the targeted attacks that we consider in this work, A_1 and A_{-1} in Eq. 2 are defined as follows (also given in Table 2).

- Positive-to-negative attacks: $A_1 = \{i : i \in Y, i \in \hat{Y}_k(S)\}, A_{-1} = \emptyset$
- Negative-to-positive attacks: $A_1 = \emptyset, A_{-1} = \{i : i \notin Y, i \notin \hat{Y}_k(S)\}$

Due to the discrete structure of the text classification problems, solving Eqs. 1 and 2 numerically is not usually possible. Instead of that, adversarial attacks on text classification problems are usually done in two steps: first, finding important words, and second, manipulating the most important words in order to reach the adversarial goals. In the subsequent

[1] In our setting, we measure similarity of S and S' by the number of different words in them, and also the similarity of their word embbedings by Universal Sentence Encoder (Cer, Yang, Kong, Hua, Limtiaco, John, Constant, Guajardo-Céspedes, Yuan, Tar, et al., 2018)

paragraphs, following Jin et al. (2020) for a black-box attack using Bert, we describe how to find important words in a sequence based on the target of the attack, and how to perform word substitution.

3.3 Finding important words

In a black-box attack, the only information available from the model is the output scores. We compute the importance of each word by masking that word in the document and measuring how much we are closer to our goal based on the target of the attack and the output scores.

Formally, assume $S = [w_1, ..., w_n]$ is a document, and $S_{\setminus w_i} = [w_1, ..., w_{i-1}, [\text{MASK}], w_{i+1}, ...]$ is the document in which the i-th word is masked. In a non-targeted attack, the importance of the i-th word is computed as follows:

$$I_{w_i} = \sum_{l \in \hat{Y}_k(S)} g_l(S) - g_l(S_{\setminus w_i}) \tag{3}$$

This equation assigns an importance score to word w_i by summing the changes in the scores of predicted labels when that word is masked.

Similarly, in positive-to-negative attacks, the important words should be those which decrease the output score of the targeted label a_1 more than other words when they are masked. Hence, the importance of the i-th word is computed as follows:

$$I^p_{w_i} = g_{a_1}(S) - g_{a_1}(S_{\setminus w_i}) \tag{4}$$

Furthermore, for negative-to-positive attacks, the importance of the i-th word is the difference in the output score of the targeted label $a_{\_1}$, after masking the i-th word:

$$I^n_{w_i} = g_{a_{\_1}}(S_{\setminus w_i}) - g_{a_{\_1}}(S) \tag{5}$$

Algorithm 1 Positive-to-negative attack pseudocode

Input: targeted label a_1, sample $S = [w_1, ..., w_n]$, score function $g(.)$, label predictor function $\hat{Y}_k(.)$, a Bert language model, maximum allowed change rate θ

Output: Adversarial sample S'

for $i = 1, ..., n$ **do**
  $S_{\setminus w_i} \leftarrow [w_1, ..., w_{i-1}, [\text{MASK}], ..., w_n]$
  $I^p_{w_i} \leftarrow g_{a_1}(S) - g_{a_1}(S_{\setminus w_i})$
end for
$I = argsort([I^p_{w_1}, ..., I^p_{w_n}])$ ▷ sorting in descending order
$t \leftarrow 1$, $S^0 \leftarrow S$
while $t < \theta$ **do**
  $i \leftarrow I[t]$
  $H \leftarrow$ suggested words by Bert for S^{t-1}, in which the i-th word is masked
  $d \leftarrow 0$
  for w_k in H **do**
    $S^{t-1}_{w^t_k} \leftarrow [w^{t-1}_1, ..., w^{t-1}_{i-1}, \{w_k\}, ..., w^{t-1}_n]$
    $d_0 \leftarrow g_{a_1}(S^{t-1}) - g_{a_1}(S^{t-1}_{w^t_k})$
    if $d_0 > d$ **then**
      $w^t \leftarrow w_k$, $d \leftarrow d_0$
    end if
  end for
  $S^t \leftarrow [w^{t-1}_1, ..., w^{t-1}_{i-1}, \{w^t\}, ..., w^{t-1}_n]$
  $S' \leftarrow S^t$, $t \leftarrow t + 1$
  if $a_1 \notin \hat{Y}_k(S')$ **then**
    return S'
  end if
end while
return None

3.4 Word substitution

Since word substitution can be the same for multiclass and multilabel problems, we can use the existing methods to replace important words. We use Bert model (Devlin, Chang, Lee, and Toutanova, 2018; Li, Ma, Guo, Xue, and Qiu, 2020) for this purpose which leads to a context-aware method and produces fluent adversarial samples. To this end, we mask the important words of a sample one by one and pass that sample to a Bert model to generate candidates for the masked words. In each trial t, we pick the word suggested by the Bert model for which the difference between the output scores towards our goal is maximized. For non-targeted and positive-to-negative attacks, this is obtained by:

Table 3 The statistics of AmazonCat-13K and Wikipedia-31K Bhatia et al. (2016). APpL and ALpP denote the average documents per label and the average labels per document, respectively

Dataset	# Training	# Test	# Labels	APpL	ALpP
AmazonCat-13K	1,186,239	306,782	13,330	448.5	5.04
Wikipedia-31K	14,146	6,616	30,938	8.5	18.6

$$w^t = \arg\max_k \sum_{j \in \Gamma} g_j(S^{t-1}) - g_j(S^{t-1}_{w^t_k}) \tag{6}$$

where S^{t-1} is the sample after changing $t-1$ important words, and $S^{t-1}_{w^t_k}$ is that sample when the t-th important word is replaced by the k-th suggested word from Bert. Also, Γ is $\hat{Y}_k(S)$ in a non-targeted attack, or limited to a_1 in a positive-to-negative attack. Moreover, for negative-to-positive attacks, w^t is computed as Eq. 6, but the sum should be multiplied by a negative sign and $\Gamma = \{a_{-1}\}$. We repeat masking the important words and feeding them to the network until the goal for the attack is reached, or we are out of the limit of the allowed number of changes. A pseudocode for positive-to-negative attacks is given in Algorithm 1.

4 Adversarial attacks on XMTC models

In this section, firstly, we perform targeted attacks on XMTC models (for the definition of the attacks, see Eq. 2 and Table 2). We show that XMTC models are vulnerable to positive-to-negative but more robust to negative-to-positive attacks. An important observation about positive-to-negative attacks is that their success rate has an imbalanced distribution, where one can successfully attack a tail label by changing only a few words in the document, while head classes are more robust to the attacks.

Secondly, to increase the robustness of tail classes against adversarial attacks, we replace the normal loss functions with the rebalanced variants (Qaraei, Schultheis, Gupta, and Babbar, 2021) in the targeted models. The results show that these loss functions can significantly improve the robustness of tail classes.

Finally, we analyse how much the positive-to-negative attacks are unnoticeable by taking into account the difference between the predicted probabilities for targeted labels as well as other labels before and after the attacks and also by employing precision and nDCG metrics to measure how much the predicted labels for the adversarial samples are similar to those of the real samples and also the true labels.

4.1 Setup

Adversarial attacks are performed on two XMTC models, AttentionXML and APLC-XLNet, trained on two extreme classification datasets, AmazonCat-13K and Wikipedia-31K (Bhatia, Dahiya, Jain, Kar, Mittal, Prabhu, and Varma, 2016). The statistics of these datasets are shown in Table 3. Similar to other datasets in XMTC, both datasets follow an extremely imbalanced distribution (Fig. 1).

We only perform positive-to-negative or negative-to-positive attacks, since these types of attacks are more practical in real-world problems than non-targeted attacks (Song, Jin, Huang, and Hu, 2018), and give us the opportunity to compare the behaviour of the models under attacking classes with different frequencies.

For each target label, we consider the samples in which the target label is among (not among) the true positive labels in positive-to-negative (negative-to-positive) attacks. We randomly draw the samples in which the target label is classified correctly. It means that the accuracy of the models on the drawn samples with respect to the target labels is always perfect in both of the attacks.

To treat labels with different frequencies equally, we partition label frequencies in different bins and draw an equal number of samples for each bin for all the experiments. We consider several consecutive frequencies as one bin, if there are at least L labels in that bin for which there is at least one correctly classified sample for each of the labels, where $L = 100$ for Wikipedia-31K and is 400 for AmazonCat-13K. We should note that, in our settings, we use top-5 as a threshold for dividing positive and negative predicted labels unless stated otherwise. The reason for using top-5 is that this is also the setting when the models are evaluated in terms of prediction accuracy in most of the work in the literature (Bhatia, Dahiya, Jain, Kar, Mittal, Prabhu, and Varma, 2016), since in most XMTC datasets, the average number of positive labels per point is logarithmic in terms of the total number of labels and is therefore low. For instance, AmazonCat-13K and Wikipedia-31K have only 5.04 and 18.64 positive labels per point on average, respectively.

To measure how much the adversarial samples are similar to the original samples, we use two criteria, which have become common in adversarial attacks on text classifiers (Ren, Deng, He, and Che, 2019; Xu & Veeramachaneni, 2021; Li, Ma, Guo, Xue, and Qiu, 2020; Garg & Ramakrishnan, 2020; Jin, Jin, Zhou, and Szolovits, 2020): (i) cosine similarity of the encoded samples using Universal Sentence Encoder (USE) (Cer, Yang, Kong, Hua, Limtiaco, John, Constant, Guajardo-Céspedes, Yuan, Tar, et al., 2018) which gives us a measure in [0, 1], (ii) change rate, which is the percentage of the words changed in a real sample to generate an adversarial sample.

4.2 General results

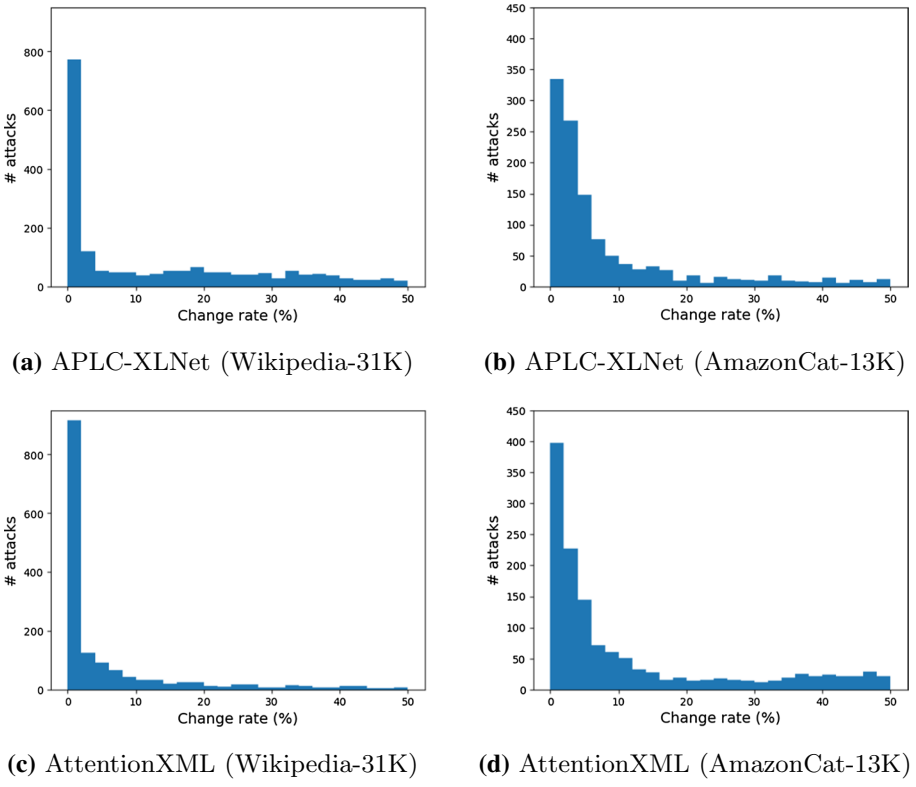
4.2.1 Positive-to-negative attacks

The results of positive-to-negative adversarial attacks on APLC-XLNet and AttentionXML for about 1000 samples uniformly drawn from different label frequency bins are shown in Table 4. Here the maximum allowed change rate is set to 10%. As the results indicate, the success rate of the positive-to-negative attacks against both models is high, which is more than 90% for Wikipedia-31K and more than 84% for AmazonCat-13K. Furthermore, the generated samples are similar to the real samples in terms of USE similarity, and the change rate is less than 2.5% in all the cases.

To measure the effect of Eq. 4 for selecting the candidates for word substitution in positive-to-negative attacks, a comparison with a positive-to-negative method in which the words to be changed are selected randomly is given in Table 4. The results show that in 3 out of 4 cases, the success rate of the attacks using Eq. 4 (indicated as WI in the table) is 17–33% higher than a random selection of the words. Also, except when APLC-XLNet is targeted on Wikipedia-31K, the similarity between adversarial and real samples is higher

Table 4 The success rate of positive-to-negative adversarial attacks against APLC-XLNet and AttentionXML on Wikipedia-31K and AmazonCat-13K. The success rate using Eq. 4 (WI) is more than 80% for all the cases with a high similarity between the adversarial and real samples and small change rate

Model	Dataset	Word selection	Success rate (%)	Similarity	Change rate (%)
APLC-XLNet	Wikipedia-31K	Random	90.12	0.76	3.10
		WI	89.94	0.74	1.14
	AmazonCat-13K	Random	61.23	0.47	3.88
		WI	85.12	0.68	1.95
Attention-XML	Wikipedia-31K	Random	78.70	0.62	1.78
		WI	96.66	0.83	0.35
	AmazonCat-13K	Random	50.93	0.39	4.42
		WI	84.40	0.67	2.20



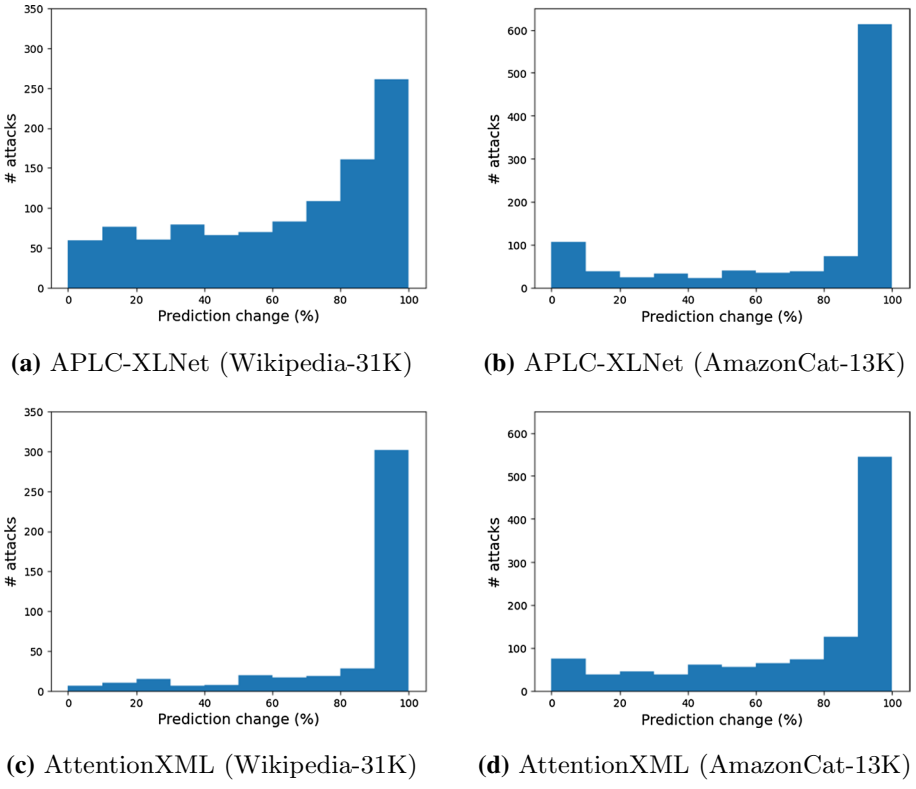
(a) APLC-XLNet (Wikipedia-31K) **(b)** APLC-XLNet (AmazonCat-13K)

(c) AttentionXML (Wikipedia-31K) **(d)** AttentionXML (AmazonCat-13K)

Fig. 2 A comparison of the change rates needed in different attacks to be successful

in WI methods. Furthermore, the change rates of WI is less than at least half of those of random word selection in all the cases.

While the results in Table 4 limit the attacks to those with change rate less than 10%, a more comprehensive analysis on the change rate is given in Fig. 2 showing that most of the attacks need a change rate in 0-10% to push the targeted label out of top-5 predictions.



(a) APLC-XLNet (Wikipedia-31K) **(b)** APLC-XLNet (AmazonCat-13K)

(c) AttentionXML (Wikipedia-31K) **(d)** AttentionXML (AmazonCat-13K)

Fig. 3 The histograms of the relative change in the predicted probabilities of targeted labels in positive-to-negative attacks, when the change rate is set to 2%. Most of the attacks achieve 80–100% decrease in the output probabilities of targeted labels by manipulating 2% of the text

Another limitation of the results given so far is that they only consider the rank of the targeted labels, and that is also limited to top-5. So if the rank of a targeted label changes by 1, from 5 to 6 for instance, that attack may be considered successful. To have an analysis independent of top-k, the reduction in the output probabilities of the targeted labels when the change rate is set to 2% (close to the change rates of the successful attacks in Table 4 for Word selection=WI) is given in Fig. 3. As the results show, most of the attacks achieve 80–100% relative decrease in the output probabilities of targeted labels. In Sect. 4.4, we show that the average prediction change of the other labels than targeted labels is significantly small meaning that the reduction of the ranks of the targeted labels can also have a big impact on the rank of those labels after the attacks.

Overall, the experiments show that XMTC models are vulnerable to positive-to-negative attacks where an adversary can fool the model not to predict a particular label by a few changes in the document.

4.2.2 Negative-to-positive attacks

While positive-to-negative attacks have high success rates on both models, having a high success rate for negative-to-positive attacks is not easy. This is due to the fact that

Table 5 Four clusters of the AmazonCat-13K and Wikipedia-31K labels. In our negative-targeted attacks, a sample may be a candidate to attack, if it has at least one positive label of those which are in the same cluster as the target label

Dataset	Clusters
AmazonCat-13K	authorship, book industry, editing, play & scriptwriting, newspapers & magazines, children's literature
	sailing, water sports, ships, canoeing, solo travel, transportation, canada
Wikipedia-31K	t-test, significance, cs483, correlation
	cmd, variable, batchfiles, command

finding the words which can increase the predicted probability of a particular label from the extremely large vocabulary space and injecting them into a document is much harder than finding the words inside a document that can lead to a lower probability for a label and replacing them with semantically similar words (positive-to-negative attacks).

To have higher success rates for negative-to-positive attacks, for each target label, we restrict the attack to the samples for which the label is close to those samples but they don't contain that label as a positive label. In our work, we assume that a label is close to a sample if that sample has at least one positive label which is in the same cluster as the target label. We perform the clustering by the balanced hierarchical binary clustering of Prabhu et al. (2018), where each label is represented by the sum of the TF-IDF representation of the documents for which that label is a positive label. Formally, assume $S_1, ..., S_N$ are our documents in the training set and $X = [x_1, ..., x_N]^T \in \mathbb{R}^{N \times V}$ are the corresponding TF-IDF representations of these documents. Also, $Z \in \{0, 1\}^{N \times L}$ consists of the one-hot labels for each document. Then $\hat{z}_l = z_l \times X$ is the representation that we use for the l-th label to perform clustering, where z_l is the l-th row of Z. Some of the clusters for AmazonCat-13K are depicted in Table 5.

After the clustering is done, for each target label $l \in C_k$ where C_k is the k-th cluster, we consider only the following samples to attack:

$$NT_l = \{S_i | i \in \{1, ..., N\}, Y(S_i) \cap C_{k \setminus l} \neq \emptyset\} \tag{7}$$

where $Y(S_i)$ consists of the indices of positive labels for the document S_i.

For negative-to-positive attacks, the number of random samples that we draw for each bin is different among different datasets and models, and it is equal to the minimum number of samples among the bins which meet the conditions.[2] Also, the cluster size is set to at least 3 labels. The results of the negative-to-positive attacks are presented in Table 6. Here the results are with and without using clustering for drawing the samples for each target label. As the results on the first row of each dataset show, the two XMTC models are robust to negative-to-positive attacks. The success rates are between 1.67% to 49.09%, while the average similarity between the real and adversarial samples cannot go above 0.2 in all the cases and is equal to 0 for the AmazonCat-13K dataset.

[2] For the negative-to-positive attacks, the conditions for picking a sample for a particular target label are: (i) the target label should not be among the positive labels of that sample, (ii) the target label should not be in the top-k predicted labels, and (iii) the sample should have at least one label inside the same cluster as the target label

Table 6 The success rate of negative-to-positive adversarial attacks against APLC-XLNet and AttentionXML on Wikipedia-31K (**W**) and AmazonCat-13K (**AC**). While the both models show robustness to the adversarial attacks, when the samples to attack are restricted to those which have at least one positive label in the same cluster as the target label, the success rate of the attack is higher than the naive case. However, all the methods show very low similarity of adversarial samples to the real samples, and also the changes rates are close

Model	Dataset	Word selection	Clustering	Success rate (%)	Similarity	Change rate (%)
APLC-XLNet	W	WI	N	49.09	0.20	3.77
		Random	Y	46.29	0.25	3.81
		WI	Y	53.70	0.30	3.39
	AC	WI	N	1.67	0.00	7.35
		Random	Y	8.14	0.05	6.67
		WI	Y	16.29	0.11	5.27
Attention-XML	W	WI	N	36.53	0.15	3.31
		Random	Y	34.00	0.18	4.31
		WI	Y	38.00	0.21	3.41
	AC	WI	N	1.95	0.00	4.39
		Random	Y	6.36	0.04	5.80
		WI	Y	12.06	0.08	5.20

Table 6 also shows the effectiveness of using clustering on increasing the success rate of the attacks and the impact of using Eq. 5 for selecting the words to be changed compared to the random baseline. Comparing rows 1 and 3 for each dataset shows clustering leads to higher success rates, ranging from 1.5% to 10%. Also, comparing rows 2 and 3 indicates higher success rates when Eq. 5 is used over the random word selection. However, all the methods have very low similarity (less than 0.3), and also the change rates are high (more than 3.3%) and close to each other. This implies the difficulty of generating negative-to-positive examples against the targeted models using the current framework, and therefore high robustness of the targeted models against these types of attacks.

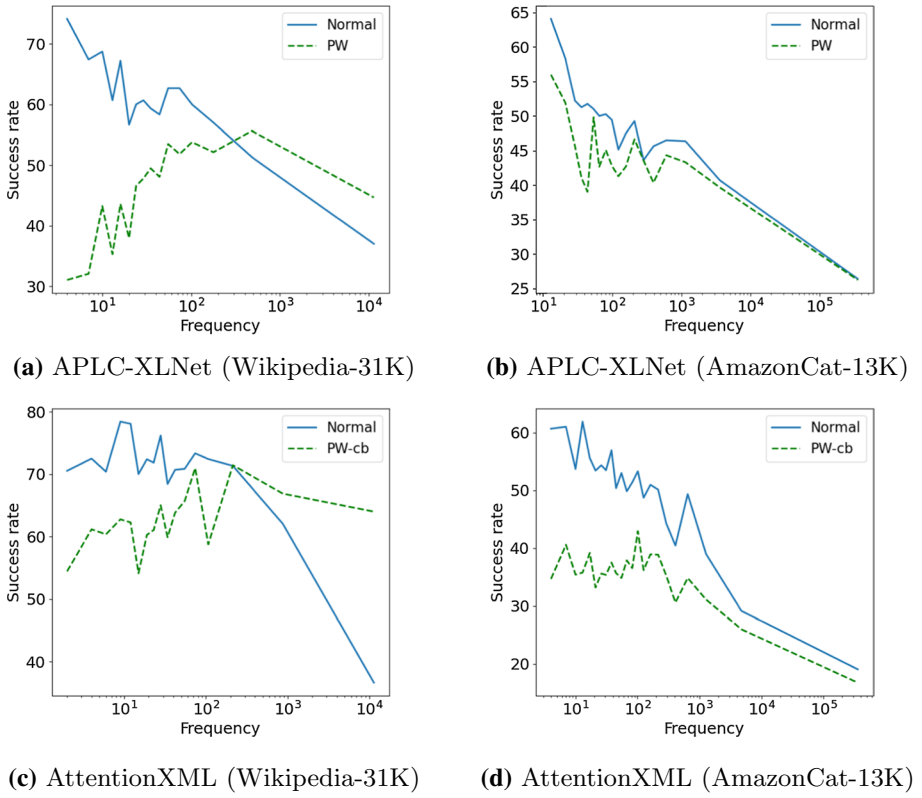
We should note that clustering of labels has been used in many XMTC works but for increasing the speed of training and evaluating the model (Prabhu, Kag, Harsola, Agrawal, and Varma, 2018; Khandagale, Xiao, and Babbar, 2020; Jiang, Wang, Sun, Yang, Zhao, and Zhuang, 2021; Mittal, Dahiya, Agrawal, Saini, Agarwal, Kar, and Varma, 2021).

4.3 Label-frequency-based results

In this subsection, first, we analyse how the success rate of the positive-to-negative attacks changes with respect to data distribution. Second, we investigate the effect of using rebalanced loss functions on this trend.

4.3.1 Attacking labels with different frequencies

The success rate of positive-to-negative adversarial attacks on labels with different frequencies are demonstrated in Fig. 4 (graphs labeled with "Normal"). We follow the setup introduced in the Setup section to categorize labels in different bins based on their frequencies, and the number of randomly drawn samples for each bin is set to 200 for



(a) APLC-XLNet (Wikipedia-31K) **(b)** APLC-XLNet (AmazonCat-13K)

(c) AttentionXML (Wikipedia-31K) **(d)** AttentionXML (AmazonCat-13K)

Fig. 4 The success rate of positive-to-negative attacks against APLC-XLNet and AttentionXML trained on two XMTC datasets for different label frequencies. An attack is successful if the similarity of the real and adversarial samples is above 0.8 and the change rate is lower than 10%. For the normal loss functions, the success rate exhibits an imbalanced behaviour, where the higher values are for the lower frequencies. Rebalanced loss functions mitigate this problem by improving the robustness of tail classes against these attacks

Wikipedia-31K and to 600 for AmazonCat-13K. Also, for these experiments, an attack is considered successful if the USE similarity of the generated adversarial samples with the real samples is above 0.8, and the change rate is less than 10%. It means that the generated adversarial samples are highly similar to the corresponding real samples.

As the figures show, the success rate of the attacks on both datasets and models has an imbalanced behaviour, where the gap between the tail and head classes is more than 30% in all the cases. It shows that it is easy to generate an adversarial sample for a tail label with a high similarity to the real samples, while this practice becomes difficult for the head classes. Some generated samples for tail classes are depicted in Table 7. While Fig. 4 shows the success rate for the case that the targeted labels are inside top-5 predicted labels, the similar trend is seen in Fig. 6 when top-3 and top-7 are used.

We would like to remind the reader that in our experiments, all the samples used for generating the adversarial attacks are classified correctly. It implies the fact that besides the challenge of predicting tail labels correctly, these labels are also more vulnerable to adversarial attacks when they are correctly predicted.

Table 7 Several adversarial samples targeted tail classes from Wikipedia-31K (W) and AmazonCat-13K (AC)

Model	Dataset	Sample	Target label	RR	RA
APLC-XLNet	W	Security-Enhanced Linux (SELinux) is a Linux system that provides a variety of security policies, including U.S. Department of Defense style mandatory access controls .	selinux	5	7
	AC	This studded protective vinyl chair Mat is top of the line .	Chair mats	5	6
	AC	Mar-Hyde One-Step Rust Converter Primer Sealer chemically reacts to ... Also proven effective on new degreased or mild iron where flash-rusting has occurred .	corrosion & Rust inhibitors	4	6
AttentionXML	W	The following are lists of notable people who intentionally terminated their own lives ... the following is a list of people whose cause of deaths is disputed or whose intention to commit suicide is doubtful .	morte	4	6
	AC	Leviton pr180-1lw decora 500w incandescent 400va passive infrared wall switch occupancy camera single pole and 3-way white .	motion-activated switches	5	6
	AC	30 built in trim kitrim for r530es- r530bs stainle ... with the acute rk51s30 stainless 30-inch trim kit .	trim kits	4	6

RR stands for the rank of targeted labels for the real samples, and RA is the rank of those labels for the adversarial samples. In all the cases, the targeted tail labels fall out of top-5 predicted labels by changing only one word

4.3.2 Robust XMTC with rebalanced losses

The rebalanced loss functions are originally proposed for the problem of missing labels and imbalanced data (Qaraei, Schultheis, Gupta, and Babbar, 2021). These loss functions, for losses which decompose over labels, such as the hinge and BCE loss, suggest the following form:

$$l(\mathrm{y}, \hat{\mathrm{y}}) = \sum_{j=1}^{L} C_j W_j l^+(y_j, \hat{y}_j) + l^-(y_j, \hat{y}_j) \tag{8}$$

where l^+ (l^-) is the positive (negative) part of the original loss. Also, C_j is a factor to rebalance the loss, and W_j is a factor to compensate for missing labels.

Suggested by Qaraei et al. (2021), we set $C_j = \frac{1-\beta}{1-\beta^{n_j}}$ (Cui, Jia, Lin, Song, and Belongie, 2019) where $\beta = 0.9$ and n_j is the number training samples for label j. Also, $W_j = 2/p_j - 1$ where p_j, called the propensity score of label j, indicates the probability of the label being present and is computed by the empirical model of Jain et al. (2016). While C_j is explicitly introduced to rebalance the loss, W_j also reweights the loss in favor of tail classes as the problem of missing labels is more pervasive in those classes.

Figure 4 demonstrates a comparison of the original models with the rebalanced variants under the adversarial attacks when the goal is to push the targeted label out of top-5 predicted labels. Here we refer to the loss modified for missing labels (only using W_j) as PW, and when the rebalancing factor (C_j) is also taken into account, we call the method PW-cb. In our experiments with rebalanced loss functions, the choice of the type of reweighting for each dataset and model depends on its prediction performance.

To compare the normal loss with the reweighted variants under the adversarial attacks, we attack the samples that are classified correctly by the normal loss, but when a reweighted loss is used to train the model.

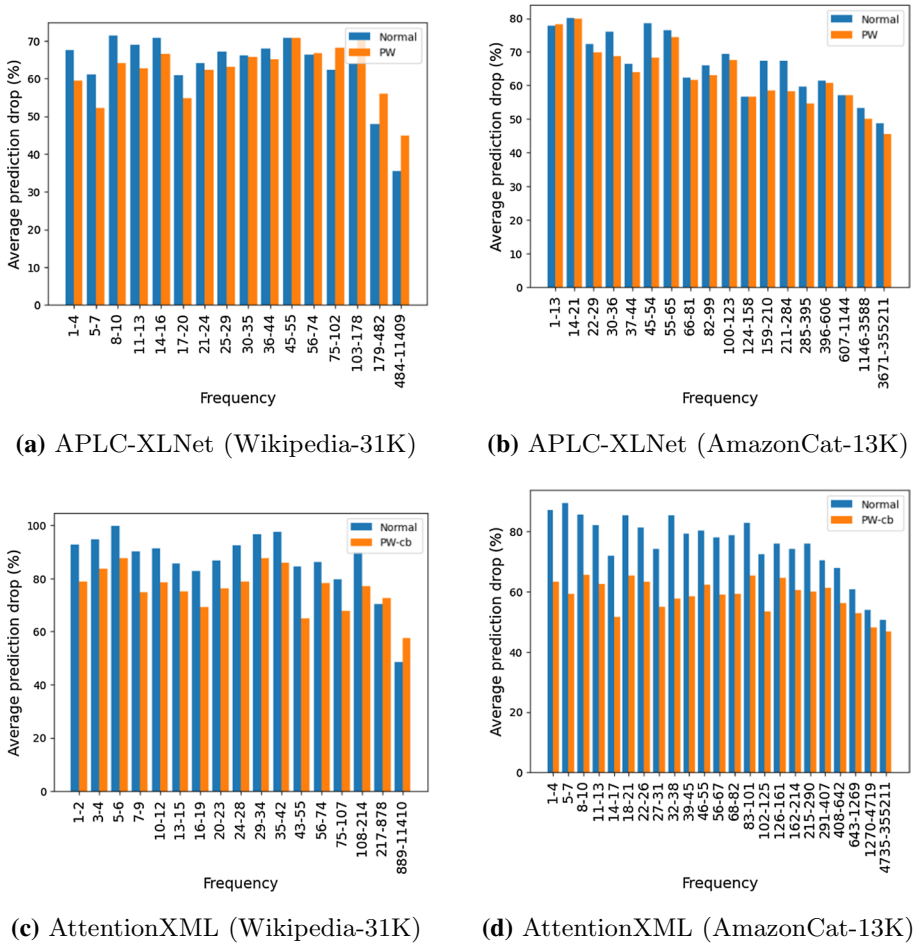
As the Fig. 4 shows, the rebalanced variants significantly improve the robustness of the models on less frequent classes. This means that using the reweighted loss functions improves the robustness of the model on the samples that are classified correctly by the normal loss but missclassified after performing the attack. The gap is large for all the model and datasets, between 10% to 40%, for the least frequent classes.

We should remark that although the reweighted loss functions improve robustness of tail classes against adversarial attacks, they have an adverse effect on head classes in Wikipedia-31K dataset. This is mostly due to two labels “wiki” and “wikipedia” which exist in more than 87% and 81% of samples, and have tiny weights in the reweighted loss functions because of having very high frequencies.

To make the analysis independent of the rank of the targeted labels, the average drop in the output probabilities of the targeted labels after the attacks are given in Fig. 5. The results show that the drop in the predicted probabilities of targeted labels in less frequent classes is smaller for rebalabnced losses on most of the cases.

4.4 Effects of positive-to-negative attacks on other labels

An ideal adversarial attack should be unnoticeable. For a multilabel problem, this unnoticeability can be measured in two ways: how much the adversarial samples are similar to the real samples, and how much the predicted labels are altered after the adversarial attacks.



(a) APLC-XLNet (Wikipedia-31K) **(b)** APLC-XLNet (AmazonCat-13K)

(c) AttentionXML (Wikipedia-31K) **(d)** AttentionXML (AmazonCat-13K)

Fig. 5 A comparison of the average drop in predicted probability of targeted labels after positive-to-negative adversarial attacks for different frequencies when a normal or rebalanced loss is used. Tail classes show less drop in predicted probabilities when a rebalanced loss is used compared to a normal loss for most of the cases

For the former, we used the change rate of the adversarial samples and also the USE similarity between the adversarial samples and the real samples in the previous subsections. For the latter, in this subsection, we measure the unnoticeability of positive-to-negative attacks by the change in the predicted probabilities of other labels than targeted labels and also the change in the rank of predicted labels after the attacks.

Table 8 shows the mean difference of the predicted probabilities of targeted labels and other labels before and after the attacks when the change rate is set to 2%. It is seen that the change in the predicted probabilities of targeted labels is at least three orders of magnitude larger than other labels.

To measure how much the predicted labels changes after adversarial attacks, we use: (i) precision, a common metric in multilabel problems which counts how many predicted labels in top-k are among true labels divided by k, (ii) overlap of the top-k predicted labels

Table 8 The mean difference between the output probabilities of targeted labels as well as other labels before and after positive-to-negative attacks when change rate is set to 2%. The mean difference between the output probabilities of other labels than targeted labels are significantly smaller than those of targeted labels

Model	Dataset	Pred. difference (targeted labels)	Pred. difference (other labels)
APLC-XLNet	Wikipedia-31K	0.27	0.00014
	AmazonCat-13K	0.55	0.00018
AttentionXML	Wikipedia-31K	0.34	0.00008
	AmazonCat-13K	0.57	0.00018

of the adversarial samples with the corresponding real samples, and (iii) normalized discounted cumulative gain (nDCG) of adversarial samples when the ground truths are the predicted labels of the corresponding real samples. This metric measures how much the rank of predicted labels of the adversarial samples are simliar to those of the real samples.

The precision metric captures the percentage of predicted labels which are among true labels for an adversarial sample. Therefore, higher precision for an adversarial sample after a successful attack indicates higher unnoticeability of the attack as the adversarial sample still has higher correlation with its true labels. However, precision may not reveal the similarity between the predicted labels of an adversarial sample and the real sample. To tackle this, we use the overlap of the predicted labels and also nDCG of the adversarial samples in which the ground truths are the predicted labels of the corresponding real samples.

To compute overlap, assuming the targeted label is inside top-k predicted labels for the real sample S and the attack is successful, the overlap of predicted labels for S' with those of S excluding the targeted label when $k > 1$ is computed as follows:

$$\text{Overlap@}k = \frac{1}{k-1} \sum_{l \in \hat{Y}_k(S')} \mathbb{I}[l \in \hat{Y}_k(S)] \tag{9}$$

where $\hat{Y}_k(S)$ and $\hat{Y}_k(S')$ are the top-k labels for a real sample S and the corresponding adversarial sample S', respectively. The -1 in the denominator is to compensate for the label which has moved out of top-k as the result of the adversarial attack.

While Overlap@k shows the similarity of top-k predicted labels for real and adversarial samples, it ignores the ranks of the labels. To this end, inspired by Brama et al. (2022), we compute nDCG over the predicted labels of the adversarial samples, where the relevance of each label for an adversarial sample is the prediction probability of that label for the corresponding real sample. We call this metric nDCG-it, which ignores the targeted label and is computed over all the labels. For an adversarial sample S', nDCG-it is computed as follows:

$$\text{nDCG-it} = \frac{1}{\text{inDCG-it}} \times \sum_{l \in \hat{Y}_L(S') \setminus t} \frac{2^{\sigma(g_l(S))} - 1}{\log(1+l)} \tag{10}$$

where $\hat{Y}_L(S')$ is the set of predicted labels sorted in descending order, t is the targeted label, and $\sigma(g_l(S))$ is the prediction probability of label l for the real sample S. Also, inDCG-it, which is the ideal nDCG-it and is therefore computed using the predicted labels for the real sample S and their prediction probabilities, is as follows:

Table 9 Precision, overlap, and nDCG-it for real and adversarial samples in positive-to-negative attacks. Precision metrics drop in adversarial samples especially for P@5 due to the fact that a successful attack has at least one irrelevant labels in top-5 predictions. However, Overlap@5 (O@5) and nDCG-it are large for all the datasets and models indicating that rank of predicted labels for adversarial samples are close to those of real samples

Model	Dataset	Sample types	P@1	P@3	P@5	O@5	nDCG-it
APLC-XLNet	Wikipedia-31K	Real	91.39	84.61	78.40	–	–
		Adv.	86.02	75.55	65.38	88.00	97.44
	AmazonCat-13K	Real	97.94	95.03	85.63	–	–
		Adv.	86.37	74.09	58.62	76.57	90.44
Attention-XML	Wikipedia-31K	Real	90.07	84.36	77.71	–	–
		Adv.	85.55	75.87	65.92	88.12	97.54
	AmazonCat-13K	Real	97.05	93.80	85.94	–	
		Adv.	86.28	75.19	61.65	80.83	93.33

$$\text{inDCG-it} = \sum_{l \in \hat{Y}_L(S) \setminus t} \frac{2^{\sigma(g_l(S))} - 1}{\log(1+l)} \tag{11}$$

nDCG-it will have a high value, if the labels with high scores for a real sample also have higher scores and therefore lower ranks in the generated adversarial sample.

Table 9 shows the results of the precision metrics on the real and adversarial samples as well as the overlap and nDCG-it. As the results show, the precision metrics are always lower for the adversarial samples especially in P@5. This is due to the fact that in successful positive-to-negative attacks, at least one true label has moved out of top-5 predicted labels. However, the overlap is more than 75% for all the adversarial samples which indicates that the top-5 predicted labels in the adversarial samples (excluding the targeted label) are highly similar to those of the real samples. Also, nDCG-it is more than 90% for all the datasets and models, which shows that the order of the predicted labels for the adversarial samples are close to those of the real samples.

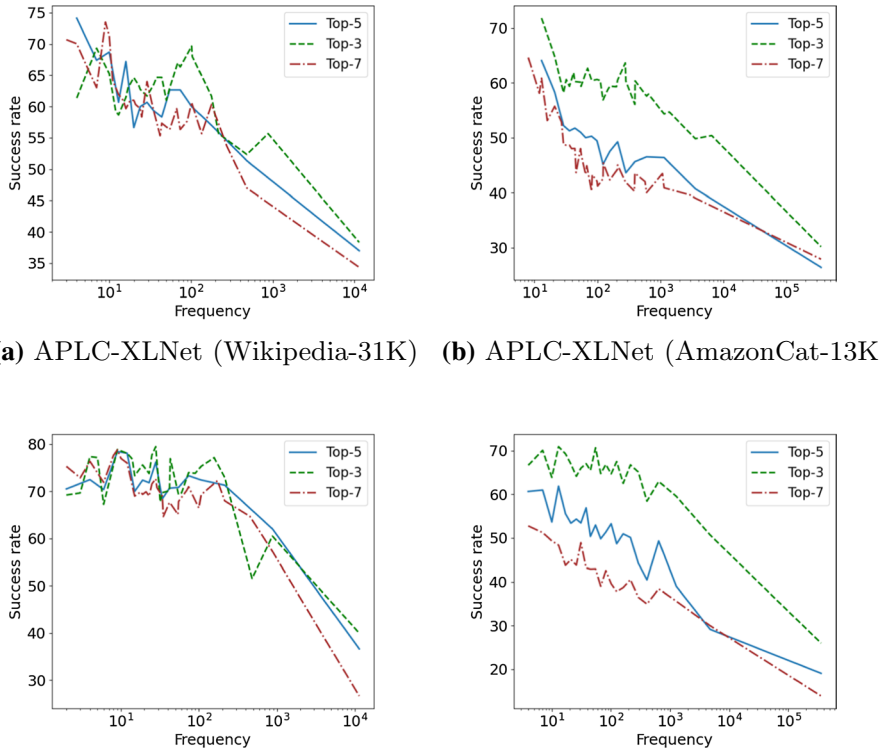
5 Conclusion

In this paper, we investigated adversarial attacks on extreme multilabel text classification (XMTC) problems. Due to the multilabel setting and extremely imbalanced data in these problems, the settings and responses for the adversarial attacks are different from the typical text classification problems. We observed that XMTC models are vulnerable to adversarial attacks when an attacker tries to remove a specific true label of a sample from the set of predicted labels, which are called positive-to-negative attacks. Also, our findings show that, besides the difficulty of correctly predicting tail classes, a new challenge in XMTC that should be considered is the low robustness of these classes against adversarial attacks. We showed that this problem can be mitigated by using the unbiased-rebalanced loss functions which reweight the loss in favour of tail classes. Two limitations in the current work which can be investigated more in the future are: 1) When the labels are pushed in or out of top-k, only top-5 is considered. The analysis could be extended to larger values for k in top-k. 2) The number of targeted labels is limited to one in each attack, while the attacks could be extended to cover multiple labels as the target labels. Also, some other remaining

questions include whether there are ways to efficiently attack XMTC models by targeting negative labels, and also how to adversarially train an XMTC model given that generating adversarial examples, which needs multiple running of the Bert model for each sample, and adding them to the clean data causes tremendous additional costs to the computationally expensive training of these models.

Results for different top-k

See Fig. (6).



(a) APLC-XLNet (Wikipedia-31K) **(b)** APLC-XLNet (AmazonCat-13K)

(c) AttentionXML (Wikipedia-31K) **(d)** AttentionXML (AmazonCat-13K)

Fig. 6 The success rate of positive-to-negative attacks for different label frequencies when the targeted label is among top-3, top-5, or top-7 predicted labels. All the results show an imbalanced success rate, where it is higher for lower frequencies and lower for higher frequencies

Author's contribution MQ did an elaborate literature review to find relevant works, designed the algorithms and implemented them through the code, ran the experiments, and wrote the first draft. RB proposed the idea, suggested frequency based analysis of adversarial examples, and the impact of clustering on the generated adversarial examples.

Funding Open Access funding provided by Aalto University. The work is funded by funding from Aalto University.

Data availability Both datasets used in our experiments are publicly available. The prepossessed datasets for APLC-XLNet can be fount at https://github.com/huiyegit/APLC_XLNet, and for attentionXML, those are located at https://github.com/yourh/AttentionXML/tree/master/data.

Code availability The code for the experiments can be found at https://github.com/xmc-aalto/adv-xmtc.

Declarations

Conflict of interest The authors have no conflicts of interests to disclose relevant to this article.

Consent for publication We give our consent for the publication of identifiable details in this paper, which can include figures, tables, and the results, in other scientific projects by mentioning the reference.

References

Agrawal, R., Gupta, A., Prabhu, Y., & Varma, M. (2013). Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 13–24.

Babbar, R., Metzig, C., Partalas, I., Gaussier, E., & Amini, M. R. (2014). On power law distributions in large-scale taxonomies. *ACM SIGKDD Explorations Newsletter, 16*(1), 47–56.

Babbar, R., & Schölkopf, B. (2017). Dismec: Distributed sparse machines for extreme multi-label classification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 721–729.

Babbar, R., & Schölkopf, B. (2019). Data scarcity, robustness and extreme multi-label classification. *Machine Learning, 108*(8), 1329–1351.

Bhatia, K., Dahiya, K., Jain, H., Kar, P., Mittal, A., Prabhu, Y., & Varma, M. (2016). The extreme classification repository: Multi-label datasets and code.

Bhatia, K., Jain, H., Kar, P., Varma, M., & Jain, P. (2015). Sparse local embeddings for extreme multi-label classification. In *NIPS*.

Brama, H., Dery, L., & Grinshpoun, T. (2022). Evaluation of neural networks defenses and attacks using ndcg and reciprocal rank metrics. http://arxiv.org/abs/2201.05071.

Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Céspedes, M., Yuan, S., & Tar, C. et al. (2018). Universal sentence encoder. http://arxiv.org/abs/1803.11175.

Cui, Y., Jia, M., Lin, T.Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277.

Devlin, J., Chang, M.W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. http://arxiv.org/abs/1810.04805.

Gao, J., Lanchantin, J., Soffa, M.L., & Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 50–56. IEEE.

Garg, S., & Ramakrishnan, G. (2020). Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6174–6181.

Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Hu, S., Ke L., Wang, X., & Lyu, S. (2021). T_kML-AP: Adversarial attacks to top-k multi-label learning. http://arxiv.org/abs/2108.00146.

Jain, H., Balasubramanian, V., Chunduri, B., Varma, M. (2019). Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 528–536.

Jain, H., Prabhu, Y., Varma, M. (2016), August. Extreme multi-label loss functions for recommendation, tagging, ranking and other missing label applications. In *KDD*.

Jiang, T., Wang, D., Sun, L., Yang, H., Zhao, Z., & Zhuang, F. (2021). Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. *In Proceedings of the AAAI Conference on Artificial Intelligence, 35*, 7987–7994.

Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020). Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *In Proceedings of the AAAI Conference on Artificial Intelligence, 34*, 8018–8025.

Khandagale, S., Xiao, H., & Babbar, R. (2020). Bonsai: diverse and shallow trees for extreme multi-label classification. *Machine Learning, 87,* 1–21.

Li, J., S. Ji, Du, T., Li, B., & Wang, T. (2018). Textbugger: Generating adversarial text against real-world applications. http://arxiv.org/abs/1812.05271.

Li, L., Ma, R., Guo, Q., Xue, X., & Qiu, X. (2020). Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6193–6202.

Medini, T. K. R., Huang, Q., Wang, Y., Mohan, V., & Shrivastava, A. (2019). Extreme classification in log memory using count-min sketch: A case study of amazon search with 50m products. *Advances in Neural Information Processing Systems, 32*, 13265–13275.

Melacci, S., Ciravegna, G., Sotgiu, A., Demontis, A., Biggio, B., Gori, M., & Roli, F. (2020). Can domain knowledge alleviate adversarial attacks in multi-label classifiers? http://arxiv.org/abs/2006.03833.

Mittal, A., Dahiya, K., Agrawal, S., Saini, D., Agarwal, S., Kar, P., & Varma, M. (2021). Decaf: Deep extreme classification with label features. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* .

Partalas, I., Kosmopoulos, A., Baskiotis, N., Artieres, T., Paliouras, G., Gaussier, E., Androutsopoulos, I., Amini, M.R., & Galinari, P. (2015). Lshtc: A benchmark for large-scale text classification. http://arxiv.org/abs/1503.08581.

Prabhu, Y., Kag, A., Harsola, S., Agrawal, R., & Varma, M. (2018). Parabel: Partitioned label trees for extreme classification with application to dynamic search advertising. In *Proceedings of the 2018 World Wide Web Conference*, pp. 993–1002.

Qaraei, M., Schultheis, E., Gupta, P., & Babbar, R. (2021). Convex surrogates for unbiased loss functions in extreme classification with missing labels. *In Proceedings of the Web Conference, 2021*, 3711–3720.

Ren, S., Deng, Y., He, K., & Che, W. (2019). Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1085–1097.

Song, Q., Jin, H., Huang, X., & Hu, X. (2018). Multi-label adversarial perturbations. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 1242–1247. IEEE.

Song, Y., Liu, Z., & Zhang, C. (2021). Multi-label text classification and text adversarial attack. In *2021 International Conference on Intelligent Computing, Automation and Applications (ICAA)*, pp. 532–536. IEEE.

Sun, L., Hashimoto, K., Yin, W., Asai, A., Li, J., Yu, P., & Xiong, C. (2020). Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert. http://arxiv.org/abs/2003.04985.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Wang, W., Xu, H., Liu, X., Li, Y., Thuraisingham, B., Tang, J. (2021). Imbalanced adversarial training with reweighting. http://arxiv.org/abs/2107.13639.

Wu, T., Liu, Z., Huang, Q., Wang, Y., & Lin, D. (2021). Adversarial robustness under long-tailed distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8659–8668.

Wu, Y., Bamman, D., & Russell, S. (2017). Adversarial training for relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1778–1783.

Xu, L., & Veeramachaneni, K. (2021). Attacking text classifiers via sentence rewriting sampler. http://arxiv.org/abs/2104.08453.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q.V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. http://arxiv.org/abs/1906.08237.

Yang, Z., Han, Y., & Zhang, X. (2020). Characterizing the evasion attackability of multi-label classifiers. http://arxiv.org/abs/2012.09427.

Ye, H., Chen, Z., Wang, D.H., & Davison, B. (2020). Pretrained generalized autoregressive model with adaptive probabilistic label clusters for extreme multi-label text classification. In *International Conference on Machine Learning*, pp. 10809–10819. PMLR.

You, R., Zhang, Z., Wang, Z., Dai, S., Mamitsuka, H., & Zhu, S. (2019). Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *NeurIPS*, pp. 5812–5822.

Zhang, W. E., Sheng, Q. Z., Alhazmi, A., & Li, C. (2020). Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST), 11*(3), 1–41.