Machine Learning (2022) 111:2515–2536
https://doi.org/10.1007/s10994-022-06176-x

# Traditional and context-specific spam detection in low resource settings

Kornraphop Kawintiranon[1] · Lisa Singh[1] · Ceren Budak[2]

Received: 1 March 2021 / Revised: 7 October 2021 / Accepted: 12 March 2022 /
Published online: 9 June 2022

## Abstract

Social media data has a mix of high and low-quality content. One form of commonly studied low-quality content is spam. Most studies assume that spam is context-neutral. We show on different Twitter data sets that context-specific spam exists and is identifiable. We then compare multiple traditional machine learning models and a neural network model that uses a pre-trained BERT language model to capture contextual features for identifying spam, both traditional and context-specific, using only content-based features. The neural network model outperforms the traditional models with an F1 score of 0.91. Because spam training data sets are notoriously imbalanced, we also investigate the impact of this imbalance and show that simple Bag-of-Words models are best with extreme imbalance, but a neural model that fine-tunes using language models from other domains significantly improves the F1 score, but not to the levels of domain-specific neural models. This suggests that the strategy employed may vary depending upon the level of imbalance in the data set, the amount of data available in a low resource setting, and the prevalence of context-specific spam vs. traditional spam. Finally, we make our data sets available for use by the research community.

**Keywords** Context-specific spam · Low-resource learning · Content-based spam detection · Cross-domain learning

Editors: João Gama, Alípio Jorge, Salvador García.

✉ Kornraphop Kawintiranon
kk1155@georgetown.edu

Lisa Singh
lisa.singh@georgetown.edu

Ceren Budak
cbudak@umich.edu

[1] Georgetown University, Washington, USA

[2] University of Michigan, Ann Arbor, USA

# 1 Introduction

Spam and junk content have been a problem on the Internet for decades (Sahami et al., 1998; Kaur et al., 2016; Ferrara et al., 2016; Dou et al., 2020). With the proliferation of direct marketing online, these forms of *pollution* continue to increase rapidly. By definition (Wu et al., 2019, https://help.twitter.com/en/rules-and-policies/twitter-rules, Pedia 2020), spam is unsolicited and unsought information, including pornography, inappropriate or nonsensical content, and commercial advertisements. These different types of spam take on new meaning in the context of social media, particularly on platforms like Twitter. For example, not everyone would view advertising as spam. When conducting a content analysis on tweets about an election, advertising about diapers is irrelevant to the election discussion and would be viewed as spam. If instead, we are analyzing content about parenting, diaper advertisements would be relevant to the content analysis and may not be viewed as spam. Because of mismatches like this, we introduce the concept of *context-specific spam* and attempt to understand how to accurately identify posts containing this form of spam, as well as more traditional forms of spam on Twitter.

Figure 1 shows the different types of content as they relate to spam. Filtering out traditional spam is an insufficient way to remove all spam tweets since context-specific spam remains. Similarly, classical spam filtering that considers all advertisements as spam is inadequate because it classifies legitimate context-specific advertising as spam. To accurately detect spam pollution, contextual understanding is required. The goal of our paper is to identify spam on Twitter, both traditional and context-specific. Researchers have been working on spam and bot detection for decades and have proposed a number of supervised learning approaches (Sahami et al., 1998; Kaur et al., 2016; Chen et al., 2015; Cresci et al., 2018). The state-of-the-art approaches extract features from both content and user information. Yet, on some platforms, user information can be difficult to obtain, either because of privacy concerns or because of API limitations. Consider the case of identifying spam among a set of tweets about a particular hashtag stream or keyword search like #metoo or #trump. Getting the user information for every

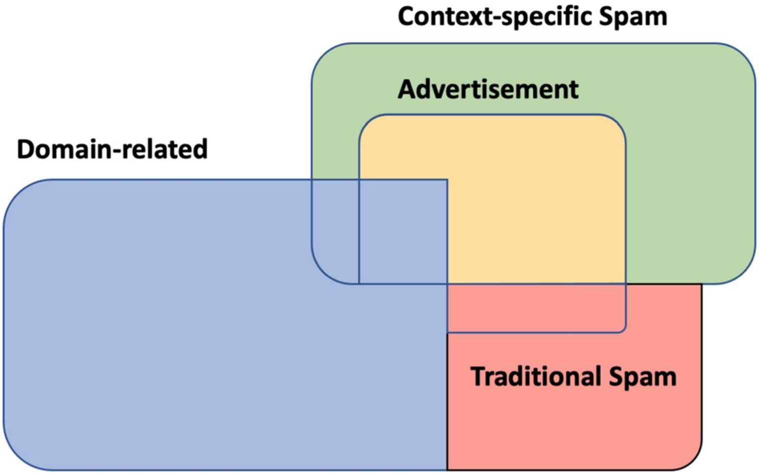


**Fig. 1** Different types of content

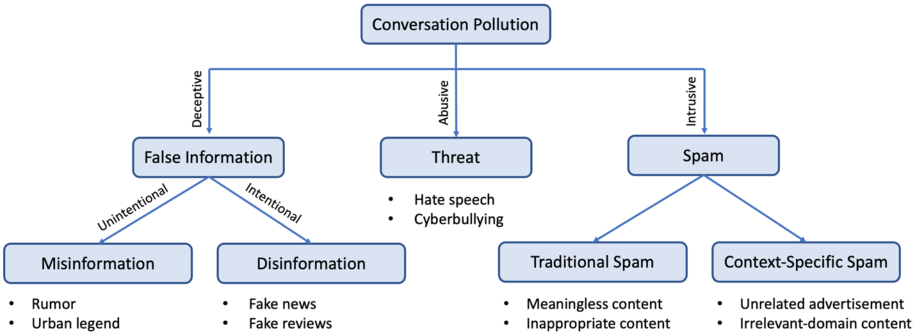


**Fig. 2** Different forms of conversation pollution

post is impractical for these frequently used hashtags that have thousands of posts daily. As such, we focus on building models that only use post content, not user information.

In this paper, our primary goal is to identify polluted information, specifically traditional spam and context-specific spam using content-based features extracted from posts. Overall, our core contributions are as follows: *(i)* we formally define different forms of conversation pollution on social media, building a useful taxonomy of poor quality information on social media; *(ii)* we present a neural network model that identifies traditional and context-specific spam in a low resource setting and show that using a language model within the neural network performs better than classic state-of-the-art machine learning models; *(iii)* we generate and make available three Mechanical Turk data sets in three different conversation domains (https://github.com/GU-DataLab/context-spam), show the existence of context-specific spam on Twitter, and how the proportion of spam varies across conversation domains; *(iv)* we demonstrate the performance impact of imbalanced training data on Twitter and show that using a neural network model is promising in this setting; and *(v)* we show that classic machine learning models are more robust to cross-domain learning when the training data are balanced, but when the training data are heavily imbalaced, a neural network with a cross-domain pre-trained language odel leads to better performance than classic models, but still not as strong performance as domain-specific training because of the presence of context-specific spam.

The remainder of this paper is organized as follows. We present our proposed conversation pollution taxonomy in Sect. 2. Next, we discuss the related work in Sect. 3. Section 4 describes our experiment design for the spam learning task. Our data sets and the labeling task are described in Sect. 5. Then, in Sect. 6, we present our empirical evaluation, followed by conclusions in Sect. 7.

## 2 Conversation pollution taxonomy

Researchers are investigating different types of conversation pollution. Kolari et al. create a taxonomy of spam across the Internet (Kolari et al., 2007). The focus of their taxonomy is on different ways to distribute spam, e.g. email, IM, blogs, etc. We present a taxonomy that groups different types of conversation pollution, where spam is one form of pollution. Figure 2 shows this taxonomy. There are three high level categories of conversation pollution: deceptive/misleading (false information), abusive/offensive

**Table 1** Examples of hypothetical tweets for the election domain

| Tweet content | Type |
|---|---|
| Smith is my favorite candidate #election2020 | Content-rich |
| Want cheap diapers for your kids? Click < URL >#election2020 | Context-specific spam |
| Order T-shirts from Alice for President Shop. Click < URL >#election2020 | Context-specific ad |
| FREE porn pics! Click < URL > #election2020 | Traditional spam |

(threat), and persuasive/enticing (spam). False information is a post containing inaccurate content, including different forms of misinformation and disinformation. Threat content is designed to be offensive and/or abusive (Wu et al., 2019). Finally, spam content attempts to persuade and entice people to click, share, or buy something. We divide spam into two categories, traditional spam and context-specific spam.

In this paper, we focus on spam-related pollution. Specifically, we introduce a new form of spam that we refer to as *context-specific* spam. Context-specific spam is any post that is undesirable given the context/theme of the discussion. This includes context-irrelevant posts like irrelevant advertising. For the purposes of this paper, we consider advertising to be posts that are intended to promote a product or service. Irrelevant advertising is advertising that is not related to the discussion domain.

More formally, let $T = \{t_1, t_2, ..., t_n\}$ be a tweet database containing $n$ tweets. Different subsets of tweets are related to different thematic domains, $D^i$ and $T = \{D^i \cup D^j\ \forall i, j\}$. Let $S$ be a set of traditional spam tweets such that $S \subset T$. While traditional spam is domain-independent, spam can be specific to a domain. We define context-specific spam $C^i$ as spam that is specific to a thematic domain $D^i$, $(C^i \subset D^i)$. An example of $C^i$ is irrelevant advertising.

To provide more insight, suppose that we are analyzing tweets about an election, i.e. the domain of conversation is elections ($D^i = election$). Table 1 shows hypothetical, example tweets. The green tweets are examples of domain-relevant tweets. Given the domain, an advertisement about diapers (row 2) is irrelevant and therefore context-specific spam ($C^i$). However, a tweet promoting a candidate's campaign (row 3) is an advertisement relevant to the domain ($C^i \cup S$) and therefore is not considered spam.

The goal of this paper is to provide researchers with automated approaches to remove irrelevant content from a particular domain of Twitter data. To that end, we propose and evaluate methods to identify $S \cup C^i$ (all spam) for different domains $D^i$ in a low resource setting, i.e. limited training data, and different levels of imbalanced training data. Both these constraints are important because of the cost associated with labeling training data and the imbalance of these training data sets with respect to spam. It is not unusual for less than 10% of the training data to be labeled as traditional spam or context-specific spam.

## 3 Related work

The definition of conversation pollution, especially on social media, is ambiguous at best. We divide this section into two subsections, focusing on different types of pollution detection. While methods for identifying content-based spam on Twitter are emerging, to the best of our knowledge, none of the state-of-the-art works include context-specific spam detection (Kaur et al., 2016; Wu et al., 2018).

### 3.1 Junk email detection

Early spam detection work focused on junk email detection (Sahami et al., 1998; Sasaki and Shinnou 2005; Wu 2009; Mantel and Jensen 2011; Cormack et al., 2007). One of the early and well-known approaches for filtering junk emails was a Bayesian model (Sahami et al., 1998) that used words, hand-crafted phrases, and the domains of senders as features. Unlike our study, these works use sender information, in addition to message content, to perform classification.

### 3.2 Spam detection on Twitter

Identifying poor quality content on social media is more challenging because the domain is broad and there are many different social media platforms with different types of posts. To further exacerbate the problem, the number of types of spam keeps increasing. Traditionally, most spammers were direct marketers trying to sell their products. More recently, researchers have identified spammers with a range of objectives (Ferrara et al., 2016; Jiang et al., 2016), including product marketing, sharing pornography, and influencing political views. These objectives vary across social media platforms. Since our work focuses on Twitter spam, we focus on literature related to spam detection on Twitter (Wu et al., 2018). We pause to mention that Twitter itself detects and blocks spam links by using Google SafeBrowsing (Chen et al., 2015), and more recently using both user and available content to identify those spreading disinformation (Safety 2020). This overall approach focuses on context-independent spam and is designed to be more global in nature. While an important step, for many public health and social science studies using Twitter data, not removing context specific spam may lead to skewed research results.

Most research focuses on detecting content polluters (Lee et al., 2011; Wu et al., 2017; El-Mawass and Alaboodi 2016; Park and Han 2016; Hu et al., 2014), i.e. individuals who are sharing poor quality content. Lee et al. (2011) studied content polluters using social honeypots and grouped content polluters into spammers and promoters. Wu et al. use discriminant analysis to identify the key post in order to identify content polluters (Wu et al., 2017). They define content polluters as fraudsters, scammers, and spammers who spread disinformation. (El-Mawass and Alaboodi (2016) used machine learning to predict Arabic content polluters on Twitter and showed that Random Forest had the highest F1-score. Our work differs from these works because we focus on the identification of pollution at the post level as opposed to polluters at the individual level. We are also interested in low resource settings where the amount of training data available is limited.

Studies focusing on spammer/bot detection tend to use both content-based and user-based information (Wu et al., 2018; Wang 2010; Mccord and Chuah 2011; Chen et al., 2015; Lin et al., 2017; Wei 2020; Hu et al., 2013; Brophy and Lowd 2020; Jeong and Kim 2018) and the best approaches achieve a precision of around 80% for training and testing on balanced datasets. Those approaches can be used when both content and user information are available. As mentioned in Sect. 1, there are scenarios where this is impractical. For example, hundreds of thousands or millions of users may post using a specific hashtag (#covid) or keyword (coronavirus), making it impractical for a researcher using those data streams to collect user information from the Twitter API. To further complicate the situation, spam is not only generated by bots. It is also produced by humans. People and companies can post advertisements or links to low-quality content. Therefore, focusing on a strategy centered on bot detection will miss some types of spam.

There are also studies on spam (as opposed to spammer) detection on Twitter (see (Kaur et al., 2016; Wu et al., 2018) for more detailed surveys). Wang proposes using a Naive Bayes model that detects spam with graph-based features (Wang 2010). The study shows that most spam tweets contain '@' or mentions and links. Since that early work, multiple studies have shown that Random Forest is effective for building models for detecting spam (Mccord and Chuah 2011; Santos et al., 2014; Chen et al., 2015; Lin et al., 2017). Chen et al., compare six algorithms that use both content-based and user-based features (Chen et al., 2015). They found that Random Forest was their best classifier, even when the training data was imbalanced. Detecting spam based solely on content is more challenging because of the lack of user information (Wang 2010). Santos et al. use traditional classifiers with Bag-of-Words (BoW) feature to detect spam using only content-based information (Santos et al., 2014). They also found that the Random Forest model outperformed other classic models. Our work differs from all this previous work since we want to detect both traditional and context-specific spam. We are also comparing classic machine learning and neural network models, conducting our analysis on three different domains on Twitter, and considering the impact of limited, imbalanced training data.

## 4 Experiment design for spam learning task

Our goal is to build a generalizable model for detecting spam on Twitter. In doing so, we investigate the following questions: (1) What are the best classic machine learning models for identifying spam on Twitter? (2) Can neural networks that incorporate a language model perform better than classic machine learning models? (3) How much does training set imbalance affect performance? (4) Are models built using one domain of Twitter training data transferable to another Twitter domain without customization? The last question is particularly important in cases when there are a small number of labels pertaining to the spam category.

Toward that end, this section describes the experimental design for understanding how different classic machine learning (Sect. 4.1) and neural network (Sect. 4.2) models perform and how transferable the models are (Sect. 4.3).

### 4.1 Classic machine learning models

We first evaluate classic machine learning models for this task given past good performance on variants of this task (Kaur et al., 2016; Wu et al., 2018; Santos et al., 2014; Hu et al., 2013, 2014). We build traditional models for each domain of interest. Figure 3 shows the details of our experimental design for generating ground truth labeled data (discussed in detail in Sect. 5), preprocessing, feature extraction, modeling and evaluation. The labeled ground truth data are inputs into the process. The data are preprocessed using simple, well-established cleaning methods.

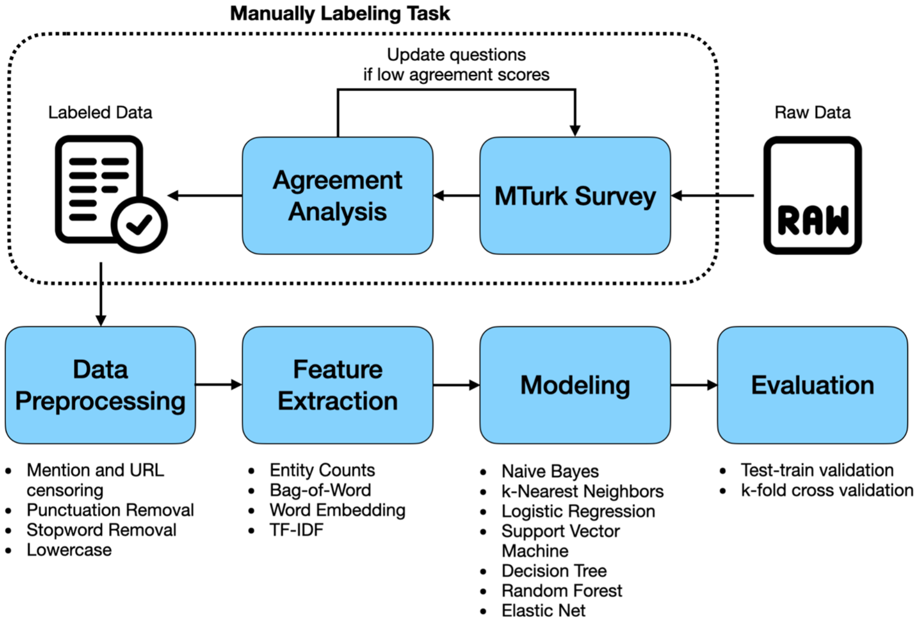


**Fig. 3** Classic machine learning models methodology overview

### 4.1.1 Feature extraction

We consider four different types of features that are widely used in spam detection research (Wu et al., 2018) including entity counts, bag-of-words (BoW), word embeddings, and TF-IDF score. Features are mixed and matched for different models (Fig. 3).

### 4.1.2 Entity count statistics

Previous spammer detection research has incorporated different entity counts statistics (Chen et al., 2015; Brophy and Lowd 2020) such as the number of retweets, the number of friends, etc. However, since our focus is content-based analysis, we build features using only the tweet content. Entity count statistics features we consider include text length, URL count, mention count, digit count, hashtag count, whether it is a retweet, and word count after URLs are removed.

### 4.1.3 Bag-of-Words (BoW)

Santos et al. (2014) show that Random Forest with BoW performed the best in content-based spam detection so we also use word frequency counts as features.

#### 4.1.4 Word embeddings

There are several embedding techniques for word representation. We use GloVe (Pennington and Socher 2014)—the most widely-used pre-trained word2vec for Twitter—to represent each word and then concatenate them as a feature vector for each sentence in a tweet.

#### 4.1.5 TF-IDF

A classic information retrieval technique for identifying important words is computing the term frequency-inverse document frequency (TF-IDF) scores. We consider a variant of the BoW model where we use the TF-IDF weight instead of the word frequency to represent the words in each tweet.

#### 4.1.6 Machine learning algorithms

We build various classic machine learning models, including Naive Bayes (NB), k-Nearest Neighbors (kNN), Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF). Given the high dimensionality and sparse feature space, we also evaluate using Elastic Net (EN) since it has been shown to work well for spammer detection by making the sparse learning more stable (Hu et al., 2013, 2014). All the models are trained using different combinations of features described above. We build and test each model for each domain.

### 4.2 Exploiting neural language models

Given the success of many neural models for text classification tasks using Twitter, we propose using a classic neural model that incorporates domain specific knowledge through the use of a language model. We hypothesize that using a language model specific to a particular domain will improve the accuracy of our spam detection models in that domain, providing necessary context. Toward that end, we incorporate a well-known neural language model, BERT (Devlin et al., 2018) and fine-tune it for this task. BERT has been used successfully for other learning tasks, including sentiment analysis (Munikar et al., 2019), natural language inference (Hossain et al., 2020) and document summarization (Cachola et al., 2020).

Our neural model begins by fine-tuning BERT to build a domain specific language model (LM) using unlabeled tweets from the domain. For example, if we are interested in gun violence, we would use a large number of tweets that discuss gun violence to fine-tune BERT. BERT uses a bidirectional transformer (Vaswani et al., 2017) and its representations are jointly conditioned on both the left and right context in all layers. The output from the language model is input into a single layer neural network for the classification task. The architecture of our neural model is shown in Fig. 4. We used the BERT tokenizer to tokenize a sentence into a list of tokens as input for BERT. After the multi-layer of transformers, we used a dropout rate of 0.1 in order to avoid over-fitting. We then fed the output vectors into a single-layer of neural network with softmax.

The classifier is a single layer neural network as shown in Eq. 1, where $y$ represents the output vector from the classifier, $W$ is a weight vector randomly initialized, $x$ represents a

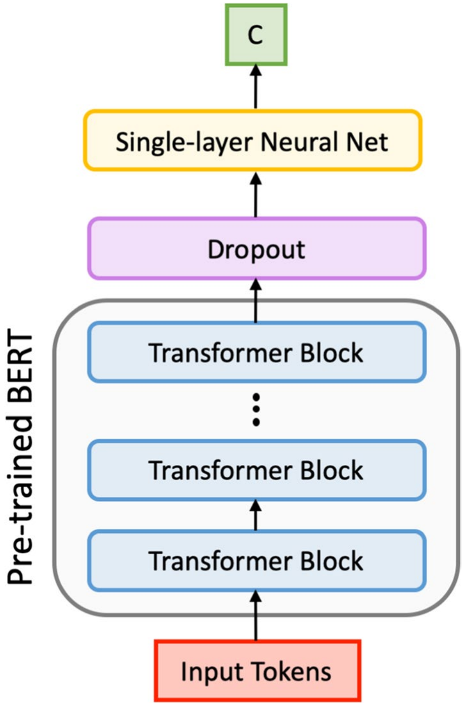


**Fig. 4** The structure of our neural network model

contextual representation vector from BERT after the dropout layer (see Fig. 4), and $b$ is a bias vector.

$$y = Wx^T + b \tag{1}$$

The weights of the classifier are updated using the cross-entropy loss function shown in Eq. 2. The class label $C$ is obtained using the softmax function (see Eq. 3) to normalize the values of the output vector $y$ from the classifier in order to obtain a probability score for each class. All BERT-based models are trained using the Adam optimizer (Kingma and Ba 2014) with learning rates of $1e-6$, $5e-6$, $1e-5$, and a fixed batch size of 32. The models are fine-tuned with a maximum epoch of 20 with early stopping. We constructed a language model for each domain and test the model in the same domain.

$$Loss(y, class) = -y[class] + \log\left(\sum_j \exp(y[j])\right) \tag{2}$$

$$C = \operatorname*{argmax}_j (\text{softmax}(y)) \tag{3}$$

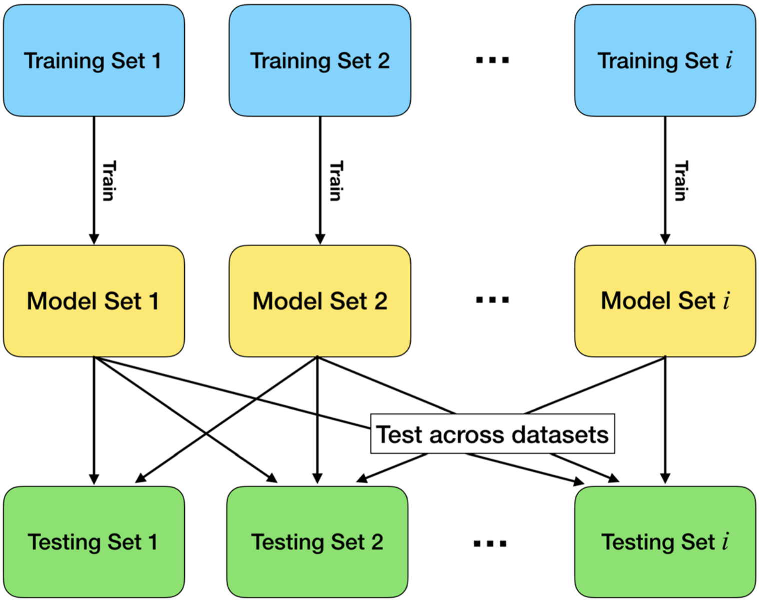


**Fig. 5** The high-level process to test model transferability

### 4.3 Model transferability

One of our goals is to understand the strengths and limitations of our models for different domains on Twitter. For example, soccer is a different domain from politics. Toward that end, we design an experiment to measure how transferable each model built in one domain is to other domains. Figure 5 shows our experimental framework. First, we train and test models within the same domain independently. We also train models in one domain and then test them across other domains to determine cross-domain generalizability and transferability of spam detection models on Twitter. In the case of the neural network model, we use a language model built across all the domains to determine its effectiveness in settings where limited ground truth data exists and the labeled ground truth data are imbalanced. We surmise that cross-domain learning will be beneficial for identifying traditional spam, but not be as accurate for context-specific spam.

## 5 Labeling spam in multiple twitter domains

The need for ground truth data is two-fold. First, we need to determine whether or not context-specific spam is present and identifiable in tweets. Second, if it is present and identifiable, we need ground truth data to help build a reliable, predictive model. Since we are interested in comparing models on different domains of Twitter data, we also collected ground truth data on more than one domain. This section begins by describing the different data sets we use for our empirical evaluation (Sect. 5.1). We then explain how we generated the ground truth data (Sect. 5.2) and the characteristics of the spam in the ground truth data for each domain (Sect. 5.3).

### 5.1 Data description

We use three different domains of Twitter data for our empirical evaluation: meToo (a social movement focused on tackling issues related to sexual harassment and sexual assault of women), gun-violence, and parenting. The meToo data set was constructed by collecting tweets that include #meToo through the Twitter API. This data set contains over 12 million tweets from October 2018 to October 2019, the first year of the larger online movement. The gun-violence data set was collected using keywords and hashtags related to gun-violence through the Twitter API. Keywords used include guns, suicide, gun deaths, etc. We used data from 2017, approximately 22 million tweets. The parenting data set was constructed by collecting the tweets of 75 authorities who primarily post and discuss parenting topics and collecting tweets using parenting-related keywords and hashtags. Example authorities include parenting magazines and medical sites. For this data set, we collected over 200 million tweets in 2019. As is evident from the descriptions, these data sets cover a wide range of topics.

### 5.2 Ground truth data collection

We collected ground truth data by setting up multiple Amazon Mechanical Turk (MTurk) tasks, one for each domain of interest. MTurk is an established approach for labeling tasks (Buhrmester et al., 2011). We ask three raters to answer three questions per tweet:

– *Question 1:* Is the tweet about domain $D^i$?
– *Question 2:* Is the statement an advertisement?
– *Question 3:* Is the statement spam?

For each question, we have definitions and examples to help raters answer the questions accurately. To make sure we separate advertising from traditional spam, in the definition of spam included for the workers, we explicitly state that advertising is not spam. We say that spam asks people for their personal information, contains harmful or inappropriate content/links including malware, phishing, or pornography, or is not understandable, e.g. not a complete sentence, not human language, etc. For this study, context-specific spam is limited to irrelevant advertising. Therefore, if Question 1 is false and Question 2 is true, we consider that context-specific spam. If Question 3 is true, we consider that traditional spam. Setting up the questions in this way allows us to see the prevalence of traditional and context-specific spam in the ground truth data sets.

From each of the data sets, we randomly sample 5000 English tweets, ensuring that there are no duplicates and that each tweet contains textual content after removing URLs. We also ensure that each tweet belongs to only one domain (either parenting, metoo or gun-violence). We recognize that tweets may belong to multiple domains. Here, we focus on single domain tweets and leave multi-domain posts for future work. Each tweet is labeled by three workers. We compute Krippendorff's alpha scores (Krippendorff 2011) and conduct agreement analysis [1] to determine the quality of the labeling for the three different-domain tasks. We compute both task-based and worker-based agreement scores and average them among tasks and workers. The task-based score looks at every tweet and every rater who labeled it. The majority vote is determined and divided by the number of answers (this is the same as the number of raters in our study). The worker-based score is

**Table 2** The agreement scores fot the MTurk labeling

| Dataset | Question | Alpha | Task-based | Worker-based |
|---|---|---|---|---|
| Parenting | *is_parenting* | 0.6809 | 0.9072 | 0.8701 |
| | *is_ad* | 0.6720 | 0.8582 | 0.8191 |
| | *is_spam* | 0.3451 | 0.9717 | 0.8571 |
| MeToo | *is_metoo* | 0.5324 | 0.8841 | 0.8830 |
| | *is_ad* | 0.4607 | 0.9545 | 0.9384 |
| | *is_spam* | 0.4155 | 0.9649 | 0.9498 |
| Gun-violence | *is_gun_violence* | 0.7124 | 0.9848 | 0.9903 |
| | *is_ad* | 0.5400 | 0.9457 | 0.9290 |
| | *is_spam* | 0.7024 | 0.8702 | 0.7884 |

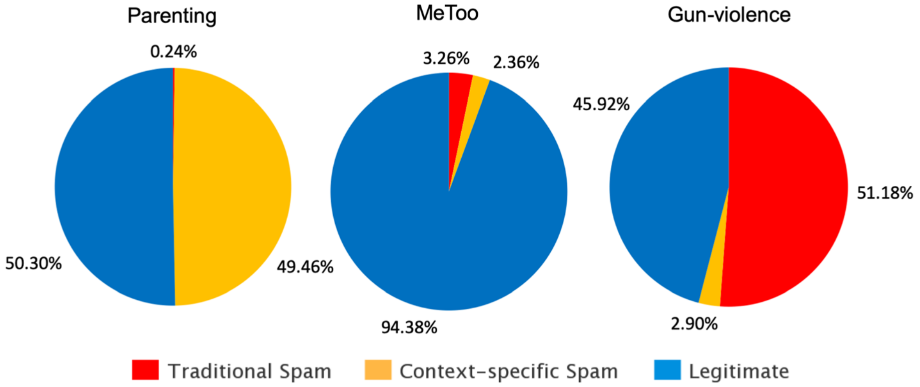


**Fig. 6** The distribution of conversation pollution in our manually annotated data

the total number of tasks a worker has a majority vote divided by the total number of tasks the worker completed. We exclude workers who complete only one task.

### 5.3 Data labeling results

Table 2 shows the scores for Krippendorff alpha, task agreement and worker agreement for each data set. By traditional standards, the alpha agreement is low (below 0.67) for five of the label categories. This results because Krippendorff's alpha relies on vote proportions and some of the label categories are heavily imbalanced. For example, in the parenting data set, there are only 12 tweets that are labeled as traditional spam and 4988 are labeled as not traditional spam. In this domain, advertising is much more prevalent, i.e. context-specific spam. These types of large imbalances reduce the overall average. Analyzing the task-based scores, we find that they are higher than 0.85 for all the questions across all three data sets. The worker-based scores vary more from 0.79 to 0.99, but in general, have a high agreement. In the few places where there is higher disagreement, the text content is more ambiguous, and workers chose the "uncertain" option sometimes. Overall, the strong agreement across both the tasks and workers give us confidence using these data to build our models.

One interesting, but unexpected finding, is the difference in the fraction of spam across domains (see Fig. 6). The parenting-domain data set is the most balanced distribution between content-rich domain tweets and spam-related pollution tweets consisting of 2,485 spam tweets (12 traditional and 2,473 context-specific spam) and 2,515 domain tweets. While at first, this may seem surprising, because some of the authorities are also traditional magazines, e.g. Parents or Baby.com, they post about products – some of which are relevant to the parenting domain, e.g. diapers, and some of which are not, e.g. iPhones. For the meToo data set, the distribution is highly imbalanced between content-rich domain tweets and spam tweets, with only 281 spam tweets (118 traditional and 163 context-specific spam) and 4,719 domain tweets. The gun-violence data set is reasonably balanced between spam and non-spam content, consisting of 2,704 spam tweets (2,559 traditional and 145 context-specific spam) and 2,296 domain tweets. The high level of traditional spam results from the high level of abusive, vulgar/inappropriate language in this data stream.

Across these data sets, we have varying distributions of spam and a varying proportion of traditional spam and context-specific spam. The parenting domain has substantially more context-specific spam when compared to the other two domains. In contrast, gun-violence has significantly more traditional spam than context-specific spam. Finally, the meToo data set has a more even distribution of context-specific and traditional spam but has a much smaller percentage of spam overall. These stark differences highlight the importance of capturing these different forms of spam. Ignoring one of them may lead to a substantial loss of information about low-quality content.

## 6 Empirical evaluation

Our empirical evaluation is organized as follows. We begin by explaining the implementation details of experiments (Sect. 6.1). Next, we present an in-domain analysis (Sect. 6.2) where models trained on data from a given domain are used to predict spam in the same domain. We also consider different levels of imbalance in the training data to determine the smallest portion of the context-specific needed in order to train a reliable classifier. Next, we perform a cross-domain analysis (Sect. 6.3), where models trained on one domain are used to predict spam in other domains. These experiments are predicting spam (traditional and context-specific) vs. no spam. We then detect traditional and context-specific spam separately using the neural network model and show that we can effectively detect both types of spam (Sect. 6.4).

### 6.1 Implementation details

We now present our implementation details. All experiments were conducted using Python 3. To accelerate the preprocessing and feature extraction process, we used Apache Spark v2.4.0 (Spark 2018), taking advantage of multiple-core processing instead of using one single-core machine. The dimensions of features vary depending upon which features were extracted (Sect. 4.1) and the training data set (Sect. 5). The sample dimension numbers[1]

[1] The number of feature dimension is higher without preprocessing.

**Table 3** Experimental results of spam pollution detection on each data set

| Training set | Feature | Best classifier | F1 (10-fold) | Accuracy | Precision | Recall | F1 (test) |
|---|---|---|---|---|---|---|---|
| Parenting | Counting | RF | 0.6634 | 0.6740 | 0.6582 | 0.7240 | 0.6895 |
| | Counting + GloVe | RF | 0.7090 | 0.7140 | 0.7004 | 0.7480 | 0.7234 |
| | Counting + BoW | RF | 0.7942 | 0.8040 | 0.7857 | 0.8360 | 0.8101 |
| | Counting + TF-IDF | RF | 0.8040 | 0.8020 | 0.7849 | 0.8320 | 0.8078 |
| | Counting + TF-IDF | NN | 0.7530 | 0.7420 | 0.7318 | 0.7640 | 0.7476 |
| | BERT-parenting | NN | **0.8237** | **0.8320** | **0.8120** | **0.8640** | **0.8372** |
| MeToo | Counting | RF | 0.6273 | 0.7679 | 0.9412 | 0.5714 | 0.7111 |
| | Counting + GloVe | LR | 0.6861 | 0.8393 | 1.0000 | 0.6786 | 0.8085 |
| | Counting + BoW | LR | 0.6945 | 0.8393 | 0.9524 | 0.7143 | 0.8163 |
| | Counting + TF-IDF | RF | 0.7603 | 0.8571 | 1.0000 | 0.7143 | 0.8333 |
| | Counting + TF-IDF | NN | 0.6274 | 0.8036 | 1.0000 | 0.6071 | 0.7556 |
| | BERT-metoo | NN | **0.7766** | **0.8929** | **1.0000** | **0.7857** | **0.8800** |
| Gun-violence | Counting | SVM | 0.7058 | 0.5020 | 0.5010 | 0.9960 | 0.6667 |
| | Counting + GloVe | SVM | 0.7058 | 0.5020 | 0.5010 | **0.9960** | 0.6667 |
| | Counting + BoW | RF | 0.8166 | 0.8060 | 0.7703 | 0.8720 | 0.8180 |
| | Counting + TF-IDF | RF | 0.8500 | 0.8620 | 0.8364 | 0.9000 | 0.8671 |
| | Counting + TF-IDF | NN | 0.8211 | 0.7780 | 0.7747 | 0.7840 | 0.7793 |
| | BERT-gun-violence | NN | **0.8669** | **0.8800** | **0.9025** | 0.8520 | **0.8765** |

The highest scores for each training data set are bolded

include 7 for *Entity Count Statistics*, 7724 for *Bag-of-Words*, 1075 to 8600[2] for *Word Embeddings* and 5221 for *TF-IDF*. Processed features are then merged onto one single machine generating the final models. We use a single machine for the final model generation because some of the machine learning algorithms are not designed for a distributed environment and it is also easier for reproducibility to use a well-known package. Specifically, we use scikit-learn v0.22.1 (Pedregosa et al., 2011) for the modeling. For the neural language models (Sect. 4.2), we implemented both the language models and neural classifiers using PyTorch v1.4.0 (Paszke et al., 2019). All the language models and classifiers were trained using a Tesla T4 GPU.

For our different learning models, we conducted a sensitivity analysis using a grid-search on influential parameters. The best parameters varied by classifier, data set, and feature sets. They include $k = \{5, 7\}$ for kNN, $C = \{1, 10\}$ for LR and SVM, $criterion = \{entropy, gini\}$ for DT, $n\_estimators = \{50, 100, 200\}$ for RF, $alpha = 0.2$ and $l1\_ratio = 0.5$ for EN and $hidden\_layer\_sizes = \{100, 200\}$ for NN. The results presented in this paper use the optimized parameters for each model.

[2] GloVe embeddings include different dimension numbers from 25 dimensions to 200 dimensions. We conducted a sensitivity analysis and present the results using the best parameters.

**Table 4** Different distributions of spam pollution in the meToo domain

| % of spam | Feature | Best classifier | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| 5vs95 | Counting | RF | 0.9676 | 0.7526 | 0.5445 | 0.6273 |
| | Counting + GloVe | LR | 0.9703 | 0.7461 | 0.6363 | 0.6861 |
| | Counting + BoW | LR | 0.9711 | 0.7618 | 0.6402 | 0.6945 |
| | Counting + TF-IDF | RF | 0.9780 | **0.8598** | 0.6874 | 0.7603 |
| | BERT-metoo | NN | **0.9788** | 0.8369 | **0.7309** | **0.7766** |
| 10vs90 | Counting | RF | 0.9495 | 0.8237 | 0.6442 | 0.7166 |
| | Counting + GloVe | LR | 0.9569 | 0.8260 | 0.7292 | 0.7701 |
| | Counting + BoW | LR | 0.9587 | 0.8094 | **0.7754** | 0.7887 |
| | Counting + TF-IDF | RF | 0.9658 | **0.9216** | 0.7223 | 0.8061 |
| | BERT-metoo | NN | **0.9680** | 0.9058 | 0.7651 | **0.8240** |
| 20vs80 | Counting | RF | 0.9196 | 0.8532 | 0.7294 | 0.7805 |
| | Counting + GloVe | LR | 0.9331 | 0.8642 | 0.7937 | 0.8248 |
| | Counting + BoW | LR | 0.9487 | 0.8821 | **0.8615** | 0.8696 |
| | Counting + TF-IDF | LR | 0.9437 | 0.8922 | 0.8185 | 0.8510 |
| | BERT-metoo | NN | **0.9502** | **0.8975** | 0.8507 | **0.8713** |

The highest scores for each split ratio are bolded

### 6.2 Within domain analysis

In this experiment, we compare classic machine learning models and a neural network model for predicting spam within a single domain, i.e. the training set and the test set are sampled from the same conversation domain. The gun-violence and parenting data sets are split into a 90/10 train/test split. Because the meToo data set only contains 281 spam tweets, we use 90% of the spam tweets in the training set (253 tweets) and have a small, balanced test set containing 28 spam and 28 domain tweets. Experiments were conducted and validated using 10-fold cross-validation on training sets and then evaluated on hold-out test sets. All baseline classifiers were constructed using different combinations of features as described in Sect. 4. The language model is pre-trained on 500,000 unlabeled domain-specific tweets.

Table 3 shows experimental results for both the training and testing data sets for detection of all spam - traditional and context-specific. Only the best classifier in each feature group is presented for ease of exposition. We also show the results of the vanilla neural network, i.e. a neural network without the pre-trained language model, to demonstrate the influence of the language model. We use the *Counting+TF-IDF* feature combination for the vanilla neural network since it tends to perform the best among four feature sets based on F1 (10-fold). The F1 scores are presented for both the training and test experiments. Accuracy, precision and recall are only shown for the test experiments. The best scores for each domain are highlighted in the table.

Focusing on the F1 scores from the 10-fold cross-validated training sets, the table shows that the neural network models with the pre-trained language models outperform all the baselines by a small amount, 1 to 2% for every domain. Similarly, focusing on the test F1 score, they outperform all the baselines by 2%, 5%, and 1% for the parenting, meToo, and gun-violence data sets, respectively. Random forest is the best classic machine learning model across domains. In general, for classic models, using the entity counting and TF-IDF

**Table 5** Comparison of 10-fold results on 80/20 imbalanced training data

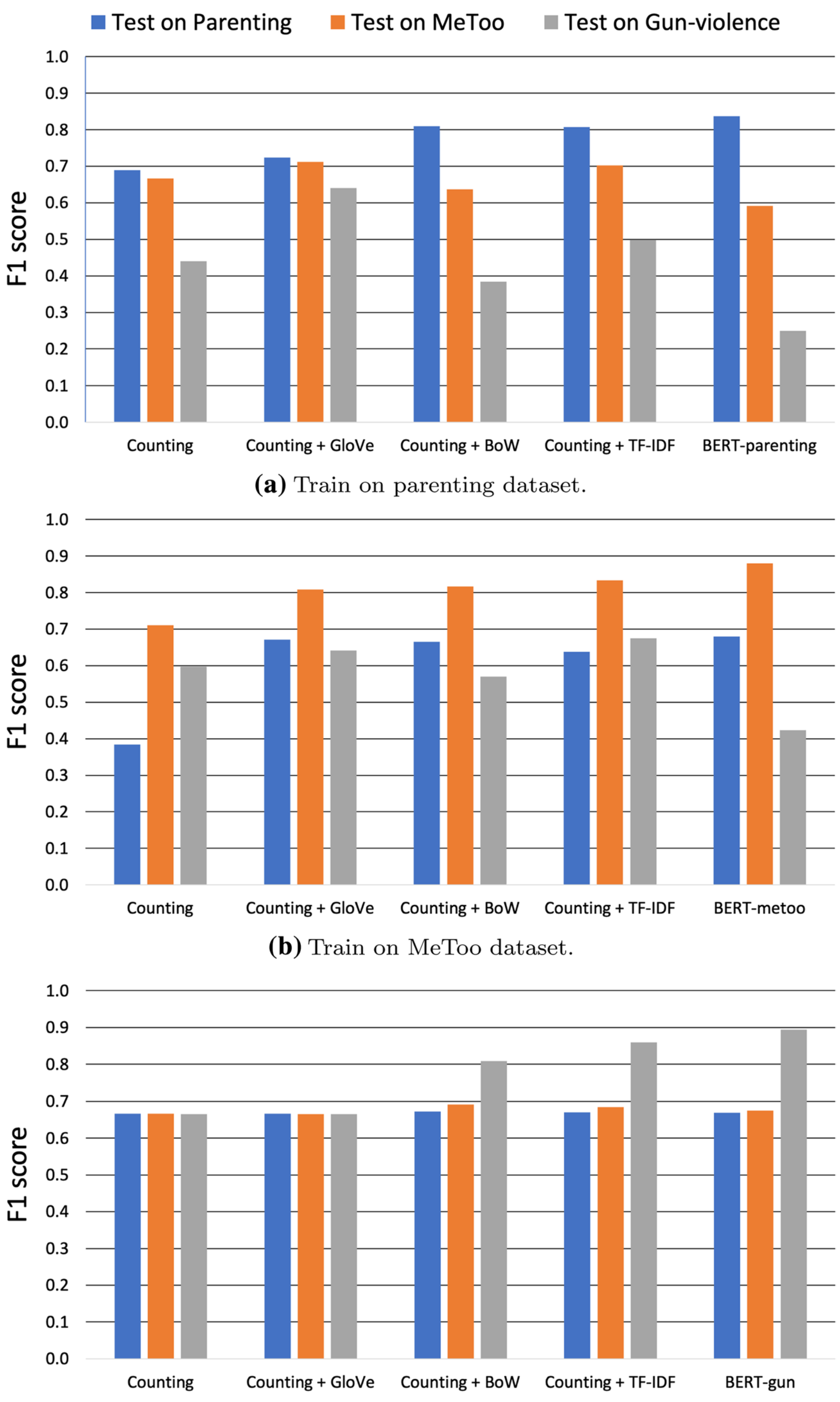
| Dataset | Feature | Best classifier | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| Parenting | Counting | DT | 0.7502 | 0.3936 | 0.4239 | 0.4022 |
| | Counting + GloVe | DT | 0.7659 | 0.4263 | 0.4558 | 0.4369 |
| | Counting + BoW | LR | 0.8477 | **0.7337** | 0.3882 | 0.5038 |
| | Counting + TF-IDF | LR | 0.8271 | 0.5843 | 0.4700 | 0.5177 |
| | BERT-parenting | NN | **0.8683** | 0.7271 | **0.5691** | **0.6310** |
| MeToo | Counting | RF | 0.9196 | 0.8532 | 0.7294 | 0.7805 |
| | Counting + GloVe | LR | 0.9331 | 0.8642 | 0.7937 | 0.8248 |
| | Counting + BoW | LR | 0.9487 | 0.8821 | **0.8615** | 0.8696 |
| | Counting + TF-IDF | LR | 0.9437 | 0.8922 | 0.8185 | 0.8510 |
| | BERT-metoo | NN | **0.9502** | **0.8975** | 0.8507 | **0.8713** |
| Gun-violence | Counting | DT | 0.6932 | 0.2426 | 0.2457 | 0.2426 |
| | Counting + GloVe | DT | 0.7024 | 0.2992 | 0.3597 | 0.3242 |
| | Counting + BoW | LR | 0.8199 | 0.5966 | 0.3240 | 0.4184 |
| | Counting + TF-IDF | LR | 0.8142 | 0.5496 | 0.4096 | 0.4681 |
| | BERT-gun-violence | NN | **0.8890** | **0.7663** | **0.6688** | **0.7010** |

The highest scores for each training data set are bolded

weighted word features performs better than other feature combinations. In all cases, entity counting statistics plus word features perform better than entity counting features alone. These findings are consistent with previous research (Santos et al., 2014; Wu et al., 2018). We see that the vanilla neural model without pre-training using a language model performs 9% to 12% worse on the test data sets, indicating the importance of pre-training.

Exploring this imbalance more, we consider the trade-off between a larger imbalance and a larger training set versus a smaller imbalance and a smaller training set. Beginning with the meToo data set, we adjust the distribution of the meToo training data to analyze different levels of imbalance. We consider three split distributions: using all the data (95/5 split), using 2529 domain tweets and 281 spam tweets (90/10 split), and using 1124 domain tweets and 281 spam tweets (80/20 split). We show the 10-fold cross-validation results for these three split distributions in Table 4. In this resource-constrained, highly imbalanced training data scenario, logistic regression and random forest still perform better than other classic machine learning models on all the different splits, while neural network models still have the highest F1 scores. Even though this is not surprising, it is still important to note that as the imbalance decreases by 5%, the F1 score increases 3 to 5%.

To see how generalizable this finding is, or if it is unique to the meToo domain, we create two data sets with an 80/20 split for the parenting and gun-violence domains. We sample 281 spam tweets and 1124 domain tweets in order to directly compare to the meToo results in Table 4. Table 5 shows the comparison of results from the 10-fold cross-validation on the imbalanced data. The first observation is that the parenting and gun-violence F1 scores are significantly lower - the lack of examples reduces the scores by 19% and 17% respectively for the top models, and has an even greater impact on many of the classic models (see 10-fold on Table 3). Similar to the previous results, we see that the neural network model still outperforms other models by 2 to 23%. These results indicate that spam within each domain has sufficient variability and that having only 20% spam in the training data may not lead to acceptable results for certain domains.



**(a)** Train on parenting dataset.

**(b)** Train on MeToo dataset.

**(c)** Train on gun-violence dataset.

**Fig. 7** F1 scores from different feature sets and the best classifiers trained and tested across different datasets

**Table 6** Comparison of F1 scores from classical models and language models trained from same-domain or combined tweets

| Train | Test | Best classic classifier | LM-same | LM-combined |
|---|---|---|---|---|
| Parenting | Parenting | 0.8101 | 0.8372 | 0.8370 |
| | MeToo | 0.7018 | 0.5909 | 0.5500 |
| | Gun-violence | 0.6398 | 0.2492 | 0.1958 |
| MeToo | Parenting | 0.6712 | 0.6791 | 0.7158 |
| | MeToo | 0.8571 | 0.8800 | 0.8333 |
| | Gun-violence | 0.6753 | 0.4237 | 0.4153 |
| Gun-violence | Parenting | 0.6721 | 0.6684 | 0.6702 |
| | MeToo | 0.6842 | 0.6747 | 0.6753 |
| | Gun-violence | 0.8493 | 0.8941 | 0.8833 |

The best performing classifier is underlined

### 6.3 Cross-domain analysis

We next determine how well models trained in one domain work in another domain for the spam detection task. Figure 7 shows the F1 score of the best classifiers trained on one domain and tested on other domains. The x-axis shows each model and the y-axis shows the F1 score. It is no surprise that training and testing using the same domain always performs better than building the model in one domain and testing it on different domains. But there are a few other interesting takeaways. First, the classic models show more robustness compared to the neural network models. While the neural network always has the highest F1 score when the test data set is within the same domain, its drop in performance on other test data sets is worse than some of the classic models. The most stable model across test sets is random forest while using Counting+GloVe, followed by Counting+TF-IDF. We hypothesize that because the neural network model is designed to incorporate context representations more deeply, that changing contexts impacts its performance more significantly. The other interesting finding is that the most robust training set is the gun violence one. This makes sense because it is the one with the highest fraction of traditional spam. In other words, cross-domain analysis is most beneficial when there is a large amount of traditional spam to learn from. Based on these two findings, we wanted to determine if we could improve the cross-domain performance of the neural model by pre-training the language model with data from all three domains.

To answer this question, we sampled 500,000 unlabeled tweets from each domain and combined them to fine-tune the language model (1.5M unlabeled tweets in total). We trained and tested the models for each domain. The results are shown in Table 6. The first two columns show the training and test data sets. The next column shows the F1 score of the best classic classifier, and the last two columns show the F1 score when the neural network model is pre-training on a language model from the same domain (LM-same) and when it is training on a combined domain language model (LM-combined). The underlined numbers are the best F1 scores using different models on the test sets. We see that the classic models are generally better when there is a difference in the domain. This is consistent with our results in the previous analysis. In all cases, a language model built using all of the domains performs worse or marginally better than pre-training a domain-specific language model. In other words, it is better to optimize for a specific domain by using

**Table 7** Comparison of the neural network models performance on different types of spam

| Data Set | Task | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|
| Parenting | Traditional Spam | – | – | – | – |
| | Context-specific Spam | 0.8249 | 0.8306 | 0.8133 | 0.8210 |
| | Pollution Spam | 0.8236 | 0.8172 | 0.8309 | 0.8237 |
| MeToo | Traditional Spam* | 0.9305 | 0.8927 | 0.7621 | 0.8129 |
| | Context-specific Spam* | 0.9656 | 0.9293 | 0.9015 | 0.9122 |
| | Pollution Spam* | 0.9502 | 0.8975 | 0.8507 | 0.8713 |
| Gun-violence | Traditional Spam | 0.8856 | 0.8911 | 0.8851 | 0.8879 |
| | Context-specific Spam* | 0.9475 | 0.8784 | 0.8767 | 0.8707 |
| | Pollution Spam | 0.8576 | 0.8834 | 0.8521 | 0.8669 |

**Table 8** 10-fold performances of our LM-combined models on spam from all domains

| Task | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Traditional Spam | 0.9085 | 0.9144 | 0.9026 | 0.9079 |
| Pollution Spam | 0.8566 | 0.8647 | 0.8479 | 0.8553 |
| Context-Specific Spam | 0.8623 | 0.8556 | 0.8533 | 0.8408 |

domain-specific language model pre-training than it is to pre-train across all the domains, assuming the size of data for pre-training is fixed. This is an unexpected finding. One would expect that pre-training across all the domains would improve the F1 scores when the training examples come from one domain and the testing examples from another. This was not the case - the variability of type of spam, i.e. the different proportions of traditional and context-specific spam (see Fig. 6) and their variation in language, impacted the overall performance. Therefore, building a general model for spam is complex. We will explore this more in the next set of experiments.

## 6.4 Traditional vs. context-specific spam

While it is important to be able to identify all spam, the last analysis highlighted the need to identify a specific type of spam. In this experiment, we analyze the ability of the neural network models to detect each type of spam separately. We only include metoo and gun-violence data in the traditional spam experiment since the parenting data contains too few traditional spam tweets. Due to the imbalanced data and to avoid overfitting on deep neural networks caused by oversampling, for data sets with less than 20% spam, we reduced the training sample size to maintain a 20–80 spam, not-spam ratio. These experiments were done using 10-fold cross-validation. Table 7 shows the performance of the neural network models on each task. Pollution spam is the combination of traditional and context-specific spam. While we also tested the classic models, we focus on the neural network models since they performed better than the classic models across all the data sets on this task. Tasks having a star next to their name indicate that we have reduced the sample size to maintain the 80-20 ratio. The results show that the neural network is able to perform well labeling context-specific and traditional spam on all of the domains with an F1 score of over 80%.

A question that arises as we look at the results is whether or not a single spam detector can be designed to identify both traditional and context-specific spam across all domains. To test this, we also compare detectors built using traditional spam from all the domains to all the spam pollution, including context-specific spam, across all domains. We combined traditional spam from all three domains then undersampled non-spam tweets from all three domains equally. As a result, the models were evaluated on training data sets containing 2689 traditional spam tweets and 2688 non-spam tweets (896 from each domain). For the pollution spam, regardless of type, we sampled 5470 spam tweets and 5469 non-spam tweets (1823 from each domain). We also sampled 2252 context-specific spam tweets combined from three different domains and 2253 non-spam tweets (751 from each domain) in order to evaluate how much the performance drops if we use one classifier for detecting context-specific spam from three different domains. We evaluated our results using 10-fold cross-validation. The performance results are shown in Table 8. We see that generalizing across traditional spam performs better (F1=0.91) than generalizing across both context-specific and traditional spam (F1=0.86). Generalizing across context-specific spam negatively impacts the overall performance (F1=0.84). There are 5 and 7% differences, which are significant for this task. Given this result and previous results, if there is sufficient spam of each type, then building models for each type individually can lead to higher F1 scores than building a model for both spam types together. However, in cases where the training data imbalances are large for the different types of spam combining them leads to reasonable results.

## 7 Conclusion

The goal of this paper is to demonstrate the existence of context-specific spam and build models to automatically identify both context-specific and traditional spam using only post data on Twitter. This is a challenging problem because: (1) what qualifies as spam varies across domains and as such, this task likely necessitates different models and new training data sets for each new domain; (2) labeling data is costly; and (3) different domains/themes of conversation on Twitter have different levels of spam, leading to different class balances in training data sets.

In this study, we define and show that context-specific spam exists on Twitter. We develop a broad conversation pollution taxonomy and place context-specific spam within that taxonomy. We experiment with different classic machine learning models and construct different types of text features introduced in previous literature to determine the best *content-based* algorithms for identifying spam within a single domain of posts and across multiple domains. We find that the best classic models are logistic regression and random forest, a finding that is consistent with previous literature. We analyze using a neural network model that incorporates a layer for a pre-trained language model and show that this model performs better than the best classic machine learning models and a basic neural network model when the training and testing data sets are within a single domain. However, when the training and test data sets are from different domains, the classic models are more robust than the neural models. Still, neither are as good as the domain-specific models because of the presence of context-specific spam. We also consider large imbalanced data sets and show that using a cross-domain pre-trained language model when the training data are small and imbalanced reduces the negative impact of the large imbalance.

Future work will look at other forms of conversation pollution to see if incorporating language models into neural networks can improve the state of the art. Other possible directions include considering related domains (e.g. the child behavior domain as it is closely related to the parenting domain), quantifying the levels of spam present on different platforms, and developing strategies to intervene and remove or mark those that are detracting from the conversation.

**Acknowledgements** This research was funded by National Science Foundation awards \#1934925 and \#1934494, and the Massive Data Institute (MDI) at Georgetown University. We would like to thank our funders, the MDI staff, and the members of the Georgetown DataLab for their support. We would also like to thank the anonymous reviewers for the detailed and thoughtful reviews.

## References

Amazon: Hit review policies. https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_HITReviewPolicies.html (2011)

Brophy, J., Lowd, D. (2020). Eggs: A flexible approach to relational modeling of social network spam. arXiv:2001.04909

Buhrmester, M., Kwang, T., & Gosling, S. D. (2011). Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science, 6*(1), 3–5.

Cachola, I., Lo, K., Cohan, A., Weld, D.S. (2020). TLDR: Extreme summarization of scientific documents. In: Findings of the Association for Computational Linguistics: EMNLP.

Chen, C., Zhang, J., Chen, X., Xiang, Y., Zhou, W. (2015). 6 million spam tweets: A large ground truth for timely twitter spam detection. In: Proceedings of the IEEE international Conference on Communications (ICC).

Cormack, G. V., Gómez Hidalgo, J. M., Sánz, E. P. (2007). Spam filtering for short messages. In: Proceedings of the ACM Conference on Conference on Information and Knowledge Management (CIKM).

Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., & Tesconi, M. (2018). Social fingerprinting: Detection of spambot groups through dna-inspired behavioral modeling. *IEEE Transactions on Dependable and Secure Computing, 15*(4), 561–576.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. preprint arXiv:1810.04805.

Dou, Y., Ma, G., Yu, P. S., & Xie, S. (2020). Robust spammer detection by nash reinforcement learning. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD).

El-Mawass, N., Alaboodi, S. (2016). Detecting arabic spammers and content polluters on twitter. In: Proceedings of the International Conference on Digital Information Processing and Communications (ICDIPC).

Ferrara, E., Varol, O., Davis, C., Menczer, F., & Flammini, A. (2016). The rise of social bots. *Communications of the ACM, 59*(7), 96–104.

Hossain, T., Logan IV, R. L., Ugarte, A., Matsubara, Y., Young, S., & Singh, S. (2020). Covidlies: Detecting covid-19 misinformation on social media. In: Proceedings of the Workshop on NLP for COVID-19 (Part 2) at EMNLP.

Hu, X., Tang, J., & Liu, H. (2014). Online social spammer detection. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).

Hu, X., Tang, J., Zhang, Y., & Liu, H. (2013). Social spammer detection in microblogging. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).

Jeong, S., Kim, C. K. (2018). Online spammer detection using user-neighbor relationship. In: Proceedings of the IEEE International Conference on Big Data (Big Data).

Jiang, M., Cui, P., & Faloutsos, C. (2016). Suspicious behavior detection: Current trends and future directions. *IEEE Intelligent Systems, 31*(1), 31–39.

Kaur, P., Singhal, A., Kaur, J. (2016). Spam detection on twitter: A survey. In: Proceedings of the IEEE International Conference on Computing for Sustainable Global Development (INDIACom).

Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980

Kolari, P., Java, A., Joshi, A., et al. (2007). Spam in blogs and social media, tutorial. In: Proceedings of the International AAAI Conference on Web and Social Media (ICWSM).

Krippendorff, K. (2011). Computing krippendorff's alpha-reliability. http://repository.upenn.edu/asc_papers/43. [Online; Retrieved 2/24/2021]

Kumar, S., Shah, N. (2018). False information on web and social media: A survey. preprint arXiv:1804.08559

Lee, K., Eoff, B. D., Caverlee, J. (2011). Seven months with the devils: A long-term study of content polluters on twitter. In: Proceedings of the International AAAI Conference on Web and Social Media (ICWSM).

Lin, G., Sun, N., Nepal, S., Zhang, J., Xiang, Y., & Hassan, H. (2017). Statistical twitter spam detection demystified: Performance, stability and scalability. *IEEE Access, 5,* 11142–11154.

Mantel, E., Jensen, S. (2011). Spam email detection based on n-grams with feature selection. US Patent 7,912,907.

Mccord, M., Chuah, M. (2011). Spam detection on twitter using traditional classifiers. In: Proceedings of the International Conference on Autonomic and Trusted Computing.

Munikar, M., Shakya, S., Shrestha, A. (2019). Fine-grained sentiment classification using bert. In: 2019 Artificial Intelligence for Transforming Business and Society (AITB), vol. 1, pp. 1–5. IEEE.

Park, B. J., & Han, J. S. (2016). Efficient decision support for detecting content polluters on social networks: An approach based on automatic knowledge acquisition from behavioral patterns. *Information Technology and Management, 17*(1), 95–105.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NeurIPS), 32,* 8026–8037.

Pedia, W.: Social Spam. (2020) https://en.wikipedia.org/wiki/Social_spam. [Online; Retrieved 2/11/20]

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn Machine learning in python. *The Journal of Machine Learning Research (JMLR), 12,* 2825–2830.

Pennington, J., Socher, R., Manning, C.D. (2014). Glove: Global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Safety, T. Disclosing networks of state-linked information operations we've removed. https://blog.twitter.com/en_us/topics/company/2020/information-operations-june-2020.html (2020). [Online; Retrieved 8/7/2020].

Sahami, M., Dumais, S., Heckerman, D., Horvitz, E. (1998). A bayesian approach to filtering junk e-mail. In: Learning for Text Categorization: Papers from the 1998 workshop, vol. 62.

Santos, I., Miñambres-Marcos, I., Laorden, C., Galán-García, P., Santamaría-Ibirika, A., Bringas, P. G. (2014). Twitter content-based spam filtering. In: International Joint Conference SOCO'13-CISIS'13-ICEUTE'13. Springer.

Sasaki, M., Shinnou, H. (2005). Spam detection using text clustering. In: Proceedings of the IEEE International Conference on Cyberworlds.

Spark, A. (2018). Apache spark. Retrieved January **17**, 2018.

Twitter: The Twitter Rules. https://help.twitter.com/en/rules-and-policies/twitter-rules/ (2020). [Online; Retrieved 2/11/2020].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 5998–6008.

Wang, A. H. (2010). Don't follow me: Spam detection in twitter. In: Proceedings of the IEEE International Conference on Security and Cryptography (SECRYPT).

Wei, F., Nguyen, U. T. (2020). Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. preprint arXiv:2002.01336.

Wu, C. H. (2009). Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. *Expert Systems with Applications, 36*(3), 4321–4330.

Wu, L., Hu, X., Morstatter, F., & Liu, H. (2017). Detecting camouflaged content polluters. In: Proceedings of the International AAAI Conference on Web and Social Media (ICWSM).

Wu, L., Morstatter, F., Carley, K. M., & Liu, H. (2019). Misinformation in social media: Definition, manipulation, and detection. *ACM SIGKDD Explorations Newsletter, 21*(2), 80–90.

Wu, T., Wen, S., Xiang, Y., & Zhou, W. (2018). Twitter spam detection: Survey of new approaches and comparative study. *Computers & Security, 76,* 265–284.