Social Media Relevance Filtering Using Perplexity-Based Positive-Unlabelled Learning

Sunghwan Mac Kim*
Lorica Health
Sydney, Australia
Mac.Kim@loricahealth.com

Stephen Wan, Cécile Paris, Andreas Duenser
CSIRO Data61
Sydney, Australia
Firstname.Lastname@data61.csiro.au

Abstract

Internet user-generated data, like Twitter, offers data scientists a public real-time data source that can provide insights, supplementing traditional data. However, identifying relevant data for such analyses can be time-consuming. In this paper, we introduce our Perplexity variant of Positive-Unlabelled Learning (PPUL) framework as a means to perform social media relevance filtering. We note that this task is particularly well suited to a PU Learning approach. We demonstrate how perplexity can identify candidate examples of the negative class, using language models. To learn such models, we experiment with both statistical methods and a Variational Autoencoder. Our PPUL method generally outperforms strong PU Learning baselines, which we demonstrate on five different data sets: the Hazardous Product Review data set, two well known social media data sets, and two real case studies in relevance filtering. All datasets have manual annotations for evaluation, and, in each case, PPUL attains state-of-the-art performance, with gains ranging from 4 to 17% improvement over competitive baselines. We show that the PPUL framework is effective when the amount of positive annotated data is small, and it is appropriate for both content that is triggered by an event and a general topic of interest.

Introduction

As a public domain data source, Twitter[1] can serve as a real-time information channel, providing insights about a number of social topics, ranging from syndromic surveillance, a form of public health monitoring (Cameron and Sparks 2015), to collecting public opinion and feedback on topics of social importance (Barwick et al. 2014). These examples are indicative of the range of scholarly disciplines that use social media data to supplement traditional sources for analysis. The latter includes official records (for example, hospital admittance) or polls (and more generally, surveys, interviews and focus groups) which can potentially be expensive and time-consuming to collect. In contrast, public social media data is often freely available and adds a real-time capability to many applications (for example, detection of suicidal ideation on social media (O'Dea et al. 2015)).

*This work was performed while the author was a Post-doctoral Fellow at the CSIRO.

[1]http://twitter.com

> @USER This is urgent there is currently a 3 storey building at church B/stop Oworoshoki Third mainland bridge which likely to collapse. (RELEVANT)
>
> @USER I would collapse (IRRELEVANT)
>
> Ashes 2015: Australia′s collapse at Trent Bridge among worst in history: England bundled out Australia for 60 ... http://t.co/t5TrhjUAU0 (IRRELEVANT)

Figure 1: Twitter posts with the ambiguous query term, *collapse*, outlining the difficulties of relevance filtering. Here the target content is emergency related (1st post), as opposed to fatigue (2nd post) or sporting metaphor (3rd post).

Collecting social media data for analysis, however, can be challenging in that one often first needs to define queries to retrieve relevant content from search web services, like that of the Twitter Application Programming Interface. Work such as (Cameron and Sparks 2015) relies on health experts to curate a list of queries, in this case related to influenza. (O'Dea et al. 2015) use a curated list of query terms based on how suicide is discussed in the vernacular. These lists are the result of careful assembly. Indeed, it can be complex to curate queries to ensure that only relevant data is collected, as described in the digital library work of (Barwick et al. 2014). Although time-consuming, many analysts employ such an approach, as these examples attest.

Figure 1 illustrates the challenges in data collection with examples of Twitter posts containing the word *collapse*, drawn from the Disaster CF20k data set (described below). The first example (relevant) describes a bridge potentially collapsing. The second example is not relevant, describing a person's fatigued state. The last example is also not relevant, as it uses *collapse* metaphorically in the jargon of sports commentary (cricket). Interestingly, the last example also contains words that are potentially disaster-related, *ashes* and *bridge*, highlighting how difficult the construction of a manual rule set might be.[2]

We introduce the problem of social media relevance filtering, whereby the aim is to help the analyst construct a

[2]Trent Bridge is the name of cricket stadium in the UK: en.wikipedia.org/wiki/Trent_Bridge

relevant data set for analysis without resorting to overly narrow queries (which is time-consuming to construct and can omit relevant content). We argue that this problem is well suited to the Positive Unlabelled Learning, or PU Learning, approach (Comité et al. 1999). When adapting PU Learning for text classification (Li and Liu 2003; Li, Liu, and Ng 2010), the authors note that PU Learning assumes that (1) obtaining *potentially related* data is trivial (e.g, in our case, with a general query term); (2) specifying examples of the positive (relevant) class is easy (e.g., here with a set of query expansion rules), but (3) representing the space of all non-relevant content is infeasible. This makes the scenario challenging; it is not trivial to exhaustively cover all the negative cases.

We introduce a new type of PU Learning, one that is able to represent a text better than the vector-space approaches used by existing methods. Specifically, we use language modelling methods to represent the positive class, noting that such models provide a richer representation of context. Our approach uses the perplexity metric to find examples that are as different as possible to the positive examples, given a language model of that class. The intuition is that non-relevant uses of a query word can be distinguished better using a representation of word context that incorporates information about word order, as opposed to a bag-of-words representation. This opens up the door for a family of PU Learning variants, which we call *perplexity-based PU Learning*, or PPUL. Exactly which language modelling method is used is a design choice for the system architect, hence our reference to PPUL as a framework. Here, we present two options: a statistical language model, to capture sequential word order, and a neural network language model (Variational Autoencoders), to encode the entire sentence.

In the remainder of this paper, we outline our approach and related PU Learning methods. Our experiments focus on three research questions: ***RQ1*** Can we verify our approach works for PU Learning?; ***RQ2*** Does PPUL work on Twitter data?; and ***RQ3*** Does PPUL work with a real (relevance filtering) task? In RQ1, we demonstrate the efficacy of PPUL by evaluating on an established PU Learning data set introduced by (Bhat and Culotta 2017), where we obtain state-of-the-art results. Our results for RQ2 demonstrate that PPUL is effective on a number of public Twitter data sets, adapted for evaluating PU Learning methods. Our results for RQ3 illustrate how the approach works in practice, with evaluations in two real social media case studies, each with manual annotations to measure performance. Finally, in our discussions, we describe how we have integrated this approach into an application workflow.

Our contributions are thus: (1) we introduce a novel PU Learning framework; (2) we obtain state-of-the-art results on the Hazardous Product Review data set; and (3) we provide a rich set of experiments and discussions outlining the scenarios in which PPUL is appropriate. We find that our approach effectively capitalises on limited positive annotated data, thus potentially making this an attractive approach for real world monitoring where the expert's time is limited and expensive. Furthermore, we find this is the case for both content triggered by events as well as general topics of interest.

PU Learning for Text Classification

In this work, we build on the PU Learning approach (Comité et al. 1999), as adapted for text classification by (Li and Liu 2003) and further explored by the same authors (Li, Liu, and Ng 2010). This latter work is interesting in its exploration of the PU Learning problem, and we base our exposition of the problem in part on (Li, Liu, and Ng 2010).

Li, Liu, and Ng note that for many real-world binary text classification problems, if one is not careful with obtaining a representative sample for the negative case, classification performance can actually be "harmful to the task". The problem arises due to covariate shift between the negative class data in the training set and the real-world data, which might be collected by a production system. Li, Liu, and Ng argue that PU Learning is a suitable approach for such situations, where it is also easier to obtain a representative sample of the positive case. They demonstrate this by discarding the negative annotations in their data sets (Reuters newsgroup data), and observe only a small drop in performance.

We note the conditions for PU Learning match our social media relevance filtering scenario. It is much easier for users of our software to say what they want to collect rather than what they do not want to collect. However, as noted above, existing PU Learning work, e.g., (Li and Liu 2003), uses a bag-of-words representation of text which cannot capture sequence information, motivating our investigation of a language modelling approach.

Preliminaries

We begin with some notation to assist with the description of our modification to PU Learning, drawing on the characterisation of the problem as outlined by (Li, Liu, and Ng 2010), and re-use their terminology.

Let $D = \{P_d \cup U\}$ denote a data set, where P_d is a set of positive samples $x_{p_d} \subseteq P_d$ (relevant instances), and U is a set of unlabelled samples. Let $U = \{P_u \cup N\}$ indicate unlabelled samples, consisting of another set of positive samples $x_{p_u} \subseteq P_u$ (relevant instances) and a set of negative samples $x_n \subseteq N$ (irrelevant instances). Let y be the binary label (tweet relevance) to be predicted, where $y = +1$ for positive samples and $y = -1$ for negative samples. The goal of (inductive) PU learning is to learn a scoring function from P_d and U that can be used to compute the likelihood of an unlabelled sample being positive ($y = +1$), allowing separation of P_u and $RN \subseteq N$. Following (Li, Liu, and Ng 2010), PU Learning can then be described as a 2-step process: (1) extract *reliable negatives* (RN) samples from U; (2) build a binary classifier from training data comprising P_d and RN.

Proposed Approach

We propose the use of language modelling methods and the perplexity score as a means of separating content that is unlike the positive class. The language model, which accounts for word selection conditioned on some context, is used to represent the positive class. Using this model, we can assign a probability to a text data instance (from a user-generated post/review), x, treated as a sequence of words, $w_1 \dots w_{|x|}$. Intuitively, perplexity, as an entropy-related measure, will

indicate how "surprised" one is, given some model, by a new unseen text. Our rationale is that, for a text in the positive class, $x_{p_d} \subseteq P_d$, perplexity will be low (unsurprising). Conversely, for a text in RN, perplexity should be high (surprise) related to unexpected word usage.

More formally, we define a function $f(x_i)$ that provides a perplexity score for each tweet $x_i \in U$ to be used for ranking. Equation 1 shows a general form for perplexity:

$$\text{perplexity}(x_i) = \frac{-1}{|x_i|} \sum_{j=1}^{|x_i|} \log p_\theta(w_j|\bullet) \tag{1}$$

where x_i is a text in the collection x, and $|x_i|$ is the number of words in x_i. θ are the parameters of a language model, and $\bullet$ is an implementation-specific representation of the context upon which w_j is conditioned.

In this paper, we describe two approaches to obtain a model for the positive class that provides the probability $p_\theta(w_j|\bullet)$. Our premise is that the probability distribution of elements in P_d is similar to the probability distribution of those positive elements in the unlabelled set, P_u. The probability distribution for elements of the negative class in the unlabelled set, N, however, will differ. Using the perplexity score, we can distinguish P_u from N. A lower perplexity score indicates that the text is well represented by the model (of the positive class). Therefore, the extracted RN from U are the ones with the highest perplexity with respect to a model derived from P_d. For all unlabelled data, we calculate and sort by perplexity. We use the top *n* samples with the highest perplexity as our reliable negatives RN, choosing $n = |P_d|$ for a balanced data set. We refer to this framework as Perplexity PU Learning, or PPUL.[3]

We define two approaches for calculating perplexity. The first uses an statistical n-gram language model to account for word order in the context preceding some word w_j, where j is a position in the sentence. The second approach utilises the entire sentence as context in assigning the probability for a word, w_j. For the latter, we employ Variational Auto Encoders (VAE). We now describe each of these in turn.

Statistical Language Modelling To demonstrate the utility of perplexity as given by a language model, we employ methods that have proven successful in Statistical Machine Translation (SMT), using a widely-used library, ***KenLM***, to obtain n-gram language models from data and calculate perplexity (Heafield 2011; Heafield et al. 2013).

The language model estimates the probability of a word w_j, given some preceding words. For example, a tri-gram language model would condition on (w_{j-1}, w_{j-2}). In the KenLM library, the probability estimate is implemented using maximum likelihood estimation with modified Kneser-Ney smoothing (Chen and Goodman 1998):

$$p_\theta(w_j|w_{j-1}, w_{j-2}) = \frac{\max(C(w_j, w_{j-1}, w_{j-2}) - D, 0)}{C(w_{j-1}, w_{j-2})} + \lambda \hat{p_\theta}(w_j|w_{j-1}) \tag{2}$$

where $C(w_i \dots w_j)$ denotes the frequency count of a word sequence as seen in the training data. λ is the interpolation weight, and D is the discount coefficient. The probability function in Equation 2 is substituted into Equation 1 to replace $p_\theta(w_j|\bullet)$ where $\bullet$ is the Markov context used in the n-gram language model.

Variational Autoencoder The statistical n-gram language model inherently captures word order in the context upon which a word is conditioned. However, it uses only the left-most context (occurring before the word in question). Alternatively, the conditioning context can occur on either side of the word in question. To model context in this manner, we use the Variational Autoencoder (VAE) method, a deep unsupervised generative model. In essence, the method involves learning to recreate the training data using a neural network. As such, the network's hidden layers provide a lower-dimensional feature representation, with parameter tuning based on reconstructing the original input data (Kingma and Welling 2013).

In this work, we use the VAE defined for document modelling (Miao, Yu, and Blunsom 2016) which is made up of an encoder (inference) and decoder (generation) network. The encoder learns the latent variables z from an input x ($q_\phi(\hat{x}|z)$), and here is implemented with a multilayer perceptron (MLP) which compresses the bag-of-words representation of a text $x_i \in x$ into a continuous distributed vector z. The decoder is used to reconstruct the input as $\hat{x}$ (as in a generative model, $p_\theta(\hat{x}|z)$). During training, the model parameters (ϕ and θ) are optimised by maximising the likelihood of the original input data being reconstructed using stochastic back-propagation.

We use the decoder as an explanatory model, as it can produce a probability distribution for words at each position of some input text, where the probability of each word w_i is conditioned on the latent variables z. We argue that the surrounding words around w_j provided as input are captured by the latent variables (as a lower-dimensional representation), providing a better representation of the context that will affect the probability of generating w_i.

More formally, the probability of selecting a word using VAE is:

$$p_\theta(w_j|z) = \frac{\texttt{exp}(-z^T R w_j + b_{w_j})}{\sum_{k=1}^{|V|} \texttt{exp}(-z^T R w_k + b_{w_k})} \tag{3}$$

where R is the word embedding matrix, and b_{w_j} is the bias term. Using VAE, $p_\theta(w_j|z)$ is substituted into Equation 1, where the latent variables z are now the contextual representation that replaces $\bullet$.

To train our VAE model, we use stochastic gradient descent with the ADAM optimiser (Kingma and Ba 2014),

[3]We note that the choice of *n* can affect performance. Here, to introduce PPUL, we choose n to create balanced data sets for training classifiers. We leave tuning of this hyperparameter to future work.

with an initial learning rate of 0.001. 500-dimensional word embeddings are randomly initialised. A two-layered MLP is employed with rectified linear activation functions for the encoder, and a 50-dimensional vector is used to represent latent variables. We use a single training sample to estimate stochastic gradients, namely a batch size of 1. This makes more frequent updates of parameters and speeds up the convergence by avoiding local optima.

Relevance Text Classification

Once the reliable negative set RN is selected, we can use any text classification method to obtain a binary relevance filter in the PPUL framework. The choice of method is not the focus in this paper, and so we report on classifiers trained using the Logistic Regression method, for ease of explanation. One can use any classifier, and we return to our use of Convolutional Neural Networks (CNN) when we discuss a deployment case.

Experimental Framework

Experiment Motivations

We now revisit the research questions that we use to structure our experiments. In RQ1, we ask whether a representation that captures sequential word order, resulting in our proposed PPUL, outperforms the bag-of-words representations used in existing PU Learning approaches. To measure this, we examine how PPUL performs on an established PU Learning data set compared to existing PU Learning approaches. We use the PU Learning data set introduced by (Bhat and Culotta 2017), focusing on user-generated reviews of hazardous products.

Although there is nothing specific in our description of PPUL to the type of text data (social media or otherwise), given that our proposed application is for social media relevance filtering, in RQ2, we ask how well PPUL works on Twitter data, which can tend to be shorter than other text types. This line of investigation is motivated by the fact that Twitter text can sometimes pose data sparsity problems for machine learning (Saif, He, and Alani 2012). We adapt two publicly available binary labelled social media data sets, and follow an established PU Learning evaluation procedure to adapt these for our purposes (Ren, Ji, and Zhang 2014).

Continuing an investigation of multiple social media data sets, our final research question (RQ3) asks: how does PPUL work with a real task requiring relevance filtering? We partner with a social media researcher from another discipline (psychology/social science) and look at the performance of PPUL for performing relevance filtering in two case studies.[4]

Hazardous Product Review PU Learning Data Set

The hazardous product review PU data set (*Product Review*) collected by (Bhat and Culotta 2017) was assembled to tackle the problem of identifying user-generated reviews indicating a product is hazardous. Such reviews might occur, for example, in online shopping platforms such as Amazon. For this task, online positive examples are available, when the hazardous product is brought to the attention of government regulators. In this data set, 2,010 incident reports on children products were extracted from a U.S. government department consumer complaints database, forming the set of positive samples. (Bhat and Culotta 2017) used Amazon product reviews (915,446) as a set of unlabelled samples. Bhat and Culotta performed a manual annotation task on a further 448 Amazon reviews for the test data. Descriptive statistics for this data set are presented in Table 1.

	Product Review
Unlabelled	915,446
Positive	2,010
Test set Positive	97
Test set Negative	351

Table 1: Summary of the Product Review data set.

Social Media Data Sets and Methodology

To test that the proposed PPUL methods will still work on social media data, which is typically shorter than other kinds of online text, we use two established Twitter data sets annotated with binary labels for evaluation. In this work, the data sets are *Disaster CF10K*[5] and *Bullying Traces*[6]. PPUL is applicable to any binary classification task, and we can think of the original positive binary class labels in each data set as surrogates for relevance (e.g., imagine the task is to collect all content related to disasters or cyberbullying).

Starting with completely annotated data provides the ground truth data for us to measure the performance of PPUL. To do this, we follow the procedure of (Ren, Ji, and Zhang 2014), replicating the PU Learning scenario by ignoring some of the labels in the training data which simulates the unlabelled data set. Ren, Ji, and Zhang control how many labels, specifically positive labels, are discarded, resulting in an evaluation framework that can reveal how a PU Learning algorithm might function in different scenarios where positive data might be more or less scarce.

We first use 90:10 split to divide the data into training and test sets, respectively. Following Ren, Ji, and Zhang, we keep 10%, 20% and 40% of the labelled positive data in the training data to form P, which would then leave the remaining 90%, 80% and 60% positive training data, respectively, to be combined with the labelled negatives N, to form the set U, where the labels (whether positive or negative) are ignored during machine learning in this framework.

We now describe the two data sets, the collection and annotation processes used to assemble them. A descriptive summary of the data sets for our PU learning experiments is shown in Table 2.

Disaster CF10K Data Set: This is a CrowdFlower data set containing 10,860 tweets which were harvested using

[4] In this paper, we report on the mechanics of relevance filtering in support of downstream research. The research findings from the analyst's perspective are outside the scope of this paper.

[5] https://data.world/crowdflower/disasters-on-social-media

[6] http://research.cs.wisc.edu/bullying

data set	Type	Size	Label	Size		
Bullying Traces	Training	2,486	Positive	56 (10%)	112 (20%)	225 (40%)
			Unlabelled	2,430	2,374	2,261
	Test	276			276	
Disaster CF10K	Training	9,774	Positive	407 (10%)	826 (20%)	1,652 (40%)
			Unlabelled	9,280	8,948	8,122
	Test	1,086			1,086	

Table 2: Summary of the social media data sets.

roughly 200 disaster-related keywords such as *ablaze* and *hailstorm*. Using crowdsourcing methods, these were then manually annotated with *Relevant* if it refers to a disaster event (a positive tweet). Otherwise, it was annotated with *Not Relevant* (a negative tweet). The original corpus contains 4,673 positive tweets and 6,187 negative tweets.

Bullying Traces Data Set: We used the *Bullying Traces* data set (version 3) which originally contained 7,321 Twitter annotations and was released in 2015 (Sui 2015). This data set was distributed only with Twitter message IDs (as Twitter terms and conditions dictate). Unfortunately, for this work, we were only able to recover the contents of 2,762 Twitter posts (the rest have presumably since been deleted and are no longer available). The original data set was assembled using the query terms: *bully*, *bullied*, *bullying*. Our version of the data set contains 701 positive (bullying) tweets and 2,061 (non-bullying) negative tweets. Here bullying refers to the report of a bullying episode, accusing someone as a bully, revealing oneself as a victim, or a cyber-bullying attack.

Social Media Case Study Data Sets

The following case study data sets allow us to gauge how our approach works in practice with social media analysis tasks. Here, we note that the first case study focuses on very specific content centred around a news event. The second represents data related to a topic without key dates associated with an event and is broader in scope. As such, the two case studies allow us to examine how PPUL works when the data set is smaller and event-driven, and what happens when the data set is larger but not event-driven.

Honey Food Quality (HFQ) Data Set: This data set was assembled in collaboration with researchers interested in using social media to study trust in food origins. Specifically, the end goal of this work was to study issues of trust regarding the food industry and its products. For the food industry, public perception and trust in food branding and labelling are crucial for the security of the industry and the security of the wider economy. In this case, Australian journalists had reported on incorrect labelling of the honey product, triggering social media discussions.[7] These discussions were the target of the data collection and, from the analyst's perspective, they may offer insights about the social perception of honey produce.

[7]https://www.abc.net.au/news/2018-09-03/capilano-and-supermarkets-accused-of-selling-\\fake-honey/10187628

In this case study, the query terms were *aushoney*, *aus honey*, *australian honey*, and *aussie honey*, given that the case was in Australia. These were used to collect data from 30 July to 17 October, 2018. We refer to this as the "raw" data set, R. The analyst defined simple rules outlining relevant content using keyword matching with terms such as *fake*, *pure*, and *counterfeit*. These rules were used to filter relevant content in R, providing the positive set P_d (recall that we noted that this is current standard practice for many data scientists to define a relevant set). The remaining data from the original queries was treated as the unlabelled set U. To evaluate performance, we created a test set, T, which was based on a sample of P_d and U. The analyst and a colleague manually annotated T for relevance. There was a strong level of agreement for this annotation task, with a Kappa of 0.89. The sets P_d and U were updated to remove any intersection with T. The updated P_d and U (disjoint with T) were then used for training.

This scenario is amenable to PU Learning because the negative examples are vast, and it is not possible to exhaustively describe these. In addition to the obvious romantic sense of *honey*, it so transpired that the current Australian TV series of *The Bachelor* had a character/participant referred to as the *Honey Badger*, illustrating how popular culture can introduce unforeseen referents, leading to ambiguity of meaning. The descriptive statistics for the Honey Food Quality (HFQ) data set are presented in Table 3.

Food Quality Concerns (FQC) Data Set: A second study with the same analyst was run looking again at the perception of the food industry. However, instead of focusing on a single product category, the focus was on the public perception of general food quality, a much broader topic. The motivation for the analyst was to help provide feedback to the food industry about relevant consumer factors, in the effort to facilitate economic growth and, eventually, improved consumer satisfaction.

The query terms used to collect data were phrases (and their variants) including *fresh food*, *fresh produce*, *food quality*, *food miles*, *food time to market*, *processed food* and *manufactured food*. The data set preparation was performed in an analogous manner to the creation of the HFQ data set. Data was collected from 24 October to 11 November, 2018, to produce the "raw" tweets, R. These were filtered using terms like *support local*, *best*, *healthy*, *safety*, *security*, and *affordable* to produce the positive set, P_d. The remaining set was treated as U. To measure performance, a test set, T, was created based on a sample of P_d and U. The set T was manually annotated by the analyst and a colleague, and there was

	HFQ	FQC
Unlabelled	770	7,677
Positive	195	5,000
Test set	400	400

Table 3: Summary of the case study data sets.

a high level of agreement in this process, with a Kappa of 0.91. P_d and U were updated to remove any overlap with T. The descriptive statistics for the Food Quality Concerns (FQC) data set are also presented in Table 3.

Prior Work and Baseline Methods

Here we outline the related prior approaches to PU Learning that we use as our baseline methods.

- **Naive Baseline+LR**: In PU Learning, the aim is to infer the subset of negative examples, N, from U. For this naive baseline, we assume $N = U$. Then using labelled data for the two classes, P_d and RN , where $RN \subset U$ and $|RN| = |P_d|$, we train a binary classifier, using a logistic regression classifier for consistency with the proposed methods (Li and Liu 2003).
- **One-class SVM** (Schölkopf et al. 2001): This approach, a standard baseline to PU Learning, couches the text classification problem as one of anomaly detection, using an SVM to model just the positive class (P_d). Test set data is then compared to this model. Those that are close to this model are treated, in our case, as relevant. Otherwise, the rest (detected anomalies) are treated as non-relevant.
- **Rocchio+EM** (Li and Liu 2003): to find RN, cosine similarity is first used to remove unlabelled content that is similar to the positive class. Rocchio classification (Rocchio 1971) is performed with the positive and remaining unlabelled set to propose the RN set, using tf-idf vector space to model context as a bag-of-words. To perform the classification, expectation maximisation (EM) (Dempster, Laird, and Rubin 1977) is used to iteratively learn a Bayesian classifier.
- **Rocchio+SVM** (Li and Liu 2003): This is similar to the previous method, Rocchio+EM, except that an SVM classifier is used instead of expectation-maximisation.[8]
- **Instance Weighting** (Elkan and Noto 2008): This approach, described in the context of relevance filtering for biomedical text, operates by automatically labelling the likely positive examples in the unlabelled data U. The approach relies on estimates of the probability of a positive label, for some text, which the authors show is directly proportional to the probability that a text is labelled. Using estimates from P_d, one can then estimate this probability. By identifying the positive data in U and removing them from the set, the remainder of U is treated as the negative labelled set, and an SVM classifier is trained.

[8]For both Rocchio+EM and Rocchio+SVM, we use the implementation by the authors at https://www.cs.uic.edu/~liub/LPULPU-download.html.

	Precision	Recall	F1
Naive Baseline + LR	0.392	0.500	0.439
OneClass SVM	0.192	0.234	0.211
Instance Weighting	0.392	0.500	0.439
Rocchio+EM	0.768	0.753	0.760
Rocchio+SVM	0.876	0.752	0.810
Feature Weighting	0.858	0.828	0.843
PPUL: LM+LR	0.705	0.814	0.756
PPUL: VAE+LR	0.837	0.881	**0.855**

Table 4: Experimental results for Product Review.

- **Feature Weighting** (Bhat and Culotta 2017): uses domain adaptation methods to modify feature weights, based on the occurrence of the features in U. Terms that are strongly indicative of the positive label in U have larger feature weights than terms that are less indicative. We note that the version of this method outlined in (Bhat and Culotta 2017) used some domain specific filtering rules for the review data set. The original version first filtered U using the star-rating metadata one finds with reviews in order to identify poor reviews which are more likely to be hazardous products. Such rules can limit the general applicability of this method. In contrast, our proposed methods do not rely on any such metadata. Our implementation of the baseline is adapted from (Bhat and Culotta 2017).[9]

Performance Metrics

We report on precision, recall, and F1 as is typical for classification performance. All the evaluation measures are calculated on the positive label, since the purpose of PU learning is to identify positive samples. Let $\hat{Y}$ refer to the set of predicted labels for the positive class, $\hat{Y}_{true}$ be the correct predictions, and Y represent the set of positive examples in the respective test sets. We use standard performance metrics:

$$\text{precision} = \frac{|\hat{Y}_{true}|}{|\hat{Y}|} \tag{4}$$

$$\text{recall} = \frac{|\hat{Y}_{true}|}{|Y|} \tag{5}$$

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6}$$

where $|Y|$ represents the size of some set, Y. The metrics all range from $[0, 1]$ and the higher the value, the better.

For the social media data sets, *Disaster CF10K* and *Bullying Traces*, since a sampling procedure is employed, we repeat each experiment three times. The reported metrics for these data sets are averaged across the repeated experiments, for which we also report standard deviation (Tables 5 and 6).

Method	Bullying Traces			Disaster CF10K		
	10% Avg F1	20% Avg F1	40% Avg F1	10% Avg F1	20% Avg F1	40% Avg F1
Naive Baseline + LR	0.129 (0.008)	0.222 (0.013)	0.370 (0.084)	0.185 (0.005)	0.303 (0.008)	0.552 (0.019)
OneClass SVM	0.084 (0.391)	0.222 (0.448)	0.370 (0.500)	0.307 (0.399)	0.477 (0.489)	0.581 (0.617)
Instance Weighting	0.391 (0.161)	0.448 (0.046)	0.500 (0.067)	0.399 (0.012)	0.489 (0.012)	0.617 (0.010)
Feature Weighting	0.632 (0.065)	0.660 (0.035)	0.704 (0.037)	0.594 (0.017)	0.643 (0.017)	0.668 (0.020)
Cosine-Rocchio EM	0.506 (0.127)	0.753 (0.064)	**0.815** (0.011)	0.509 (0.009)	0.622 (0.020)	0.715 (0.020)
Cosine-Rocchio SVM	0.230 (0.083)	0.538 (0.051)	0.722 (0.035)	0.341 (0.005)	0.523 (0.030)	0.672 (0.020)
PPUL: 2-gram LM + LR	**0.765** (0.010)	**0.786** (0.004)	0.787 (0.014)	0.685 (0.004)	0.683 (0.006)	0.707 (0.011)
PPUL: VAE + LR	0.740 (0.045)	0.777 (0.048)	0.782 (0.013)	**0.688** (0.009)	**0.712** (0.005)	**0.717** (0.011)

Table 5: Performance on the social media data sets.

Experimental Results

RQ1: Validating PPUL with PU Learning data

We start by showing how the PPUL variants perform on the *Product Review* data set (Table 4). The competitive baselines are Rocchio+EM, Rocchio+SVM and Feature Weighting. Of the baselines, Feature Weighting performs the best. However, we note that the implementation of Rocchio+EM and Rocchio+SVM had memory issues with the full data set, and that we were forced to use a sample of $10,000$ instances of the unlabelled data set, which may degrade performance.[10]

Our PPUL variants performs favourably, with the PPUL:VAE+LR variant outperforming all baselines. This demonstrates that the proposed method is suited to the PU Learning problem, as represented by this data set. Notably, although the Feature Weighting method was introduced with respect to this data set, our PPUL:VAE+LR method outperforms it. Recall that the Feature Weighting baseline relies on the unlabelled data having metadata to identify good and poor reviews (the stars), making the applicability of the approach limited in the general case. In contrast, our methods do not rely on any such metadata.

RQ2: Measuring performance on social media data

Having demonstrated in RQ1 that PPUL approach is indeed suitable for PU Learning, we evaluated its performance on established labelled social media (Twitter) data sets, which have been adapted for evaluating PU Learning (as described above). Table 5 shows the experimental results measured by average F1 (with standard deviations in parenthesis) for the two data sets; *Disaster CF10K* and *Bullying Traces*.

To begin with, we notice that the PU Learning approaches (Instance Weighting, Feature Weighting, Rocchio+EM, Rocchio+SVM and our two PPUL approaches) all outperform a naive treatment which is to use all the unlabelled data as negatives and to use OneClass SVM. From this, we conclude that, at least for Twitter data, some partitioning of the unlabelled data using PU Learning can improve downstream relevance filtering.

In both data sets, we observe that, in almost all cases, one of the PPUL variants is consistently superior to all the baselines on the task of relevant tweet identification. The exception is the *Bullying Traces* case using 40% of the positive data for training. In this case, the Rocchio+EM method slightly outperforms the PPUL methods by approximately 2 points. Indeed, this is a strong baseline, the third-best approach for all other testing conditions. In addition, the Rocchio methods have the ability to represent context, albeit in a more rudimentary form than our proposed methods (i.e., as a bag-of-words, whereas we use language modelling). That being said, it edges ahead in only one of the six conditions.

In Table 6, we present additional detail on two of these conditions. The table shows the average precision, recall and F1 scores for the Rocchio+EM baseline and best PPUL system for both Twitter data sets, with training on 40% of the positive data. We note that the PPUL systems favour recall, whereas the Rocchio+EM baseline favours precision.

In general, with more positive data, all PU learning approaches do better. Returning to Table 5, what is interesting is how the margin between the PPUL variants and the best baseline approach changes as the size of P changes. For the *Bullying Traces*, at 10%, the best baseline is the Feature Weighting baseline, which PPUL outperforms with a gain of 13%. This margin drops when using more positive data (20%) to approximately 4% improvement by PPUL:LM+LR over the Rocchio+EM approach (the best baseline at 20%) for the *Bullying Traces*. Similar trends can be seen in results for the Disaster CF10k data set. This highlights the strength of the PPUL approach in efficiently capitalising on limited positive annotated data, potentially making this an attractive approach for applications where the expert's time for defining or annotating positive data is limited and expensive, as in our context.

Finally, we note that when the data set is smaller in size, the statistical language model variant of PPUL may be better, as in the case of the *Bullying Traces* data set. For a larger data set, like *Disaster CF10K* (and indeed the *Product Review* data set which is larger again), the VAE+LR variant of PPUL works best. We suspect that the VAE approach starts to show its value only when there is sufficient data for the neural network approach to perform training adequately.

RQ3: Performance on Two Application Scenarios

Table 7 presents the performance of the different PU Learning approaches on our case study data sets. Again, we see that it is a PPUL variant that performs best, both in the

[9] https://github.com/tapilab/icwsm-2017-recalls

[10] Given that this distributed implementation of (Li and Liu 2003) is a binary executable, we were unable to identify if this was a theoretical or an implementation limitation.

Methods	Bullying Traces (40%)			Disaster CF10K (40%)		
	Prec	Recall	F1	Prec	Recall	F1
Rocchio+EM	0.761 (0.022)	0.879 (0.003)	0.815 (0.011)	0.857 (0.006)	0.613 (0.027)	0.715 (0.020)
Best PPUL	0.654 (0.018)	0.991 (0.003)	0.787 (0.014)	0.664 (0.012)	0.780 (0.011)	0.717 (0.011)

Table 6: Average precision, recall and F1 scores for the Bullying Traces and Disaster CF10K sets (40% of positive data).

Methods	FQC			HFQ		
	Prec	Recall	F1	Prec	Recall	F1
Naive Baseline + LR	0.68	0.878	0.766	0.974	0.342	0.507
OneClass SVM	0.682	0.845	0.755	0.982	0.252	0.401
Instance Weighting	0.665	0.878	0.757	0.971	0.459	0.624
Feature Weighting	0.689	0.873	0.77	0.965	0.369	0.534
Cosine-Rocchio EM	0.667	0.958	0.786	0.976	0.559	0.711
Cosine-Rocchio SVM	0.674	0.920	0.778	0.963	0.351	0.632
PPUL: LM+LR	0.708	0.944	**0.809**	0.904	0.595	0.717
PPUL: VAE+LR	0.537	1.000	0.698	0.632	0.968	**0.765**

Table 7: Performance on the case studies: Food Quality Concerns (FQC) and the Honey Food Quality (HFQ) data sets

case of an event-driven data set (HFQ) and a non-event related data set (FQC). In the case of the HFQ data set, the PPUL:VAE+LR method works the best (F1: 0.765) with a 0.05 (5 F1 point) lead over the Rocchio+EM baseline. For the FQC data set, the PPUL:LM+LR approach is the best, with an F1 of 0.809, outperforming Rocchio+EM by 0.02. We note that the best PPUL variant for either data set provides a reasonable balance of precision and recall as shown by the superior F1 performance: while the approach favours recall, over half the suggested positives are correct.

This may be important for real-world applications, when the target content corresponds to only a small volume of data. For these scenarios, it is important to collect as much data as possible and err on the side of including false positives (that is, preferring recall over precision). If the precision is reasonable, we can rely on an analyst to decide on which data to include in downstream analyses.

Discussion

Our experiments show that the PPUL Framework performs well in a variety of situations, as represented by the different data sets. To see why the approach works, we visualise the average perplexities for the positive (P), reliable negative (RN), unlabelled (U) and annotated negative (N) sets as box plots in Figure 2 on the Disaster CF10K data set. Recall that we use perplexity to define these sets, where perplexity represents how closely the text in each set matches the labelled positive set (P). For this graph, we used 20% of P to build the language model (with the remaining 80% added to U, as described in the evaluation section). As Table 5 showed that the PPUL:VAE+LR method was best, we use VAE for the language model here. That is, we use the VAE that was trained on P to encode text from these sets, from which the perplexity for that text can be calculated. This leads to our labels: *VAE-P*, *VAE-N*, *VAE-U*, and *VAE-RN* corresponding to P, N, U and RN, respectively.

The box plot shows that, of course, P has the lowest perplexity; we are least surprised by this text given that the model was based on P. Notably, the ground-truth negative set (N) has the same perplexity range as our inferred reliable negative set (RN), providing some validation that our inference of the set RN is performing as intended. Finally, the unlabelled data set (U) sits between P and N, as expected. This graph then shows that the sets are indeed delineated in a way matching our motivations for PPUL.

In Table 8, we present some examples of correctly and incorrectly classified posts from the FQC case studies, selected to show content including the term "quality". The correctly labelled content shows relevant tweets for content to do with food. The two examples present both a positive and negative perspective on food quality. The correctly determined non-relevant content also looks reasonable, catching off-topic content about the textile industry and art. In the interests of transparency, we also show incorrectly labelled content. The two examples of incorrectly labelled relevant content are about jobs in restaurants. This a tricky case since this is still related to the food industry. One potential way to address this is to increase the size of RN to better model non-relevant content. It is difficult to postulate why the last two examples were incorrectly labelled as non-relevant. We note that both examples have repeated punctuation which suggests that perhaps our text preprocessing before machine learning may need refinements. However, on the whole, the results in Table 7 were encouraging, with the PPUL:LM+LR method achieving an F1 of 0.809.

Deployment Sketch In this section, we briefly sketch how our PPUL framework is being coupled within a larger social media monitoring system *Vizie* (Wan and Paris 2014; Wan, Paris, and Georgakopoulos 2015), which is an analyst-in-the-loop system. The analyst first curates queries in the tool, allowing the analyst to group related queries together in a monitoring activity. For example, all variants of the *honey* queries (HFQ data set) were grouped in the same activity.

The *Vizie* system provides feedback to the analyst via dashboard on the kind of data that would be retrieved from

Test tweets from FQC	Gold label	PPUL
Correctly Labelled		
The food stuff is my favorite because it's usually super high **quality** stuff id normally not buy	RELEVANT	RELEVANT
ordered food home delivery.... best example for downgrading **quality** of food.... even street side food is much better than this.... this is happening twice....	RELEVANT	RELEVANT
textile producers from generations, is drawing deeply on new technologies to produce sturdy and high **quality** textiles. Read More: #textiles #technology #fabrics #producers #obradors #highquality #weaves #looms #traditional #tech	NON-RELEVANT	NON-RELEVANT
Me: I'm really tired ?? Also me: I have to stay up and bust my ass to produce high **quality** art to try and impress people who don't/won't like me	NON-RELEVANT	NON-RELEVANT
Incorrectly Labelled		
APD's flagship fishing vessel Markit8 got into some **quality** lingcod today down on the Lost Coast. Come in this week and enjoy the freshest Fish and Chips and Fish Sandwiches in town!! Help...	NON-RELEVANT	RELEVANT
This thread.... I've seen this happen over and over. The average consumer in the grocery store knows nothing of **quality**, and because they fall for cosmetics and marketing over quality we are all stuck with horrible produce.	NON-RELEVANT	RELEVANT
Flexible Hours - Chefs: Are you a talented cook who is as passionate about providing high **quality** food as we are about providing high quality care? Do you love cooking for people? Looking for flexible hours and to work?...	RELEVANT	NON-RELEVANT
JOB: Fort Lauderdale FL USA - Restaurant GM Restaurant Assistant GM Restaurant Manager - Ensure a high q: Ensure a high **quality** of ingredients and food preparation Create and adjust staff schedules to Ability .. #JOBS #POMPANO BEACH FLORIDA	RELEVANT	NON-RELEVANT

Table 8: Examples of tweets correctly and incorrectly classified by *PPUL* from the FQC data set.

social media platforms given his/her queries. This dashboard analyses a real-time sample of data collected by the query, generates lists of top hashtags and keywords, provides a topic clustering of salient content and shows links to matching pages in Wikipedia and Wiktionary for the query. The aim of this dashboard is to show content that indicates if the query is ambiguous and needs to be refined further.

As the system is designed to support constant monitoring of data collection quality, the analyst can activate the PPUL relevance filter if the manual rules look like they have poor coverage. The rules specify the positive set, and PPUL is triggered using cloud-based computing resources. The saved model file is then used to create the social media relevance filtering service. *Vizie* is built using the streaming platform framework Kafka[11] to manage data queue processing, the classifier is set up as a data consumer on the pipeline for content collected via the queries. Notably, the analyst only triggers this capability per monitoring activity as needed.

Future Work Currently, *Vizie* does not provide any option to iteratively gauge if the PPUL filter is performing as per the user's satisfaction. We are currently designing a workflow management system to help the analyst manage the process of classifier refinement through either retraining with annotated data or further refinement of rule sets.

For clarity and simplicity, we introduced and demonstrated the effectiveness of PPUL against state-of-the-art baselines using two instances of this framework that rely on a logistic regression classifier. We can use any text classification method in practice, and our current deployment uses a CNN text classifier (Kim 2014). There are further variants of the PPUL framework that should improve performance. For example, ensemble methods for neural network text classification have been shown to be an effective method in boosting performance. Similarly, variants of PU Learning can be subtly different in their focus, differing in whether to remove reliable positives from U or just select an RN set (as we do). Combined approaches may be worth investigating.[12]

Finally, recent work in developing neural network language models (for example, BERT (Devlin et al. 2018), ELMo (Peters et al. 2018) and ULMFit (Howard and Ruder 2018)) may further boost performance. Of these, BERT has been demonstrated to be the best on a suite of NLP tasks (Devlin et al. 2018). However, upon a preliminary investigation, we note that the default BERT encoder does not produce probability estimates of the input text, making it unsuit-

[11] https://kafka.apache.org/

[12] We thank anonymous reviewer R3 for this suggestion.

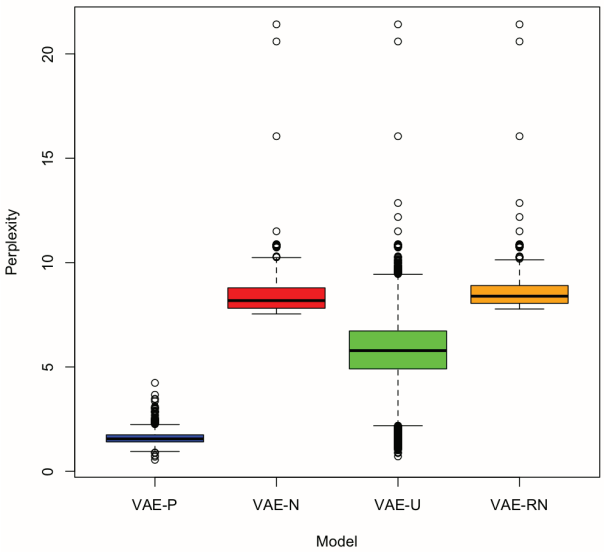

Figure 2: Boxplots showing the perplexity distributions produced by VAE on Disaster CF10K over (with 20% of positive samples for training). X-axis represents four sample types; positive (VAE-P), negative (VAE-N), unlabelled (VAE-U) and reliable negative (VAE-RN) samples, and Y-axis represents their perplexity scores.

able for calculating perplexity without some modification.[13] Furthermore, our goal in this paper is to demonstrate the utility of the PPUL framework, in which one can substitute different language modelling methods. For these reasons, we leave an exploration of models like BERT to future work.

Related Work

Relevant Tweet Identification: Keyword matching has been widely applied to relevant tweet identification (Broniatowski, Paul, and Dredze 2013; Bommannavar, Lin, and Rajaraman 2016). While simple, this approach is far from being perfect for the task (Maynard and Funk 2012; Kim, Wan, and Paris 2016). For this reason, some previous work has attempted to build traditional classifiers for various topics, formulated as a binary text classification task, e.g., television shows (Erdmann et al. 2013), cyberbullying (Xu et al. 2012), news category (Krestel et al. 2015), disaster (Stowe et al. 2016) and sentiment (Zhang and Lan 2016). It has been shown that a machine learning classifier identifies more relevant tweets than keyword matching pertaining to a particular event or topic (Neubig, Mori, and Mizukami 2013; To et al. 2017). However, previous studies used a naive process to select negative samples: a set of randomly selected tweets or a set of randomly selected tweets that do not match keywords; these may be subject to covariate shifts as indicated by (Li, Liu, and Ng 2010).

PU learning in NLP: PU learning has been studied in the context of natural language processing (NLP) (Sriphaew, Takamura, and Okumura 2009; Delort, Arunasalam, and Paris 2011; Shen, Bunescu, and Mihalcea 2012). PU learning has been used recently for various tasks, such as: identifying keyphrases from online documents (Sterckx et al. 2016), important sentences in news (Yang, Bao, and Nenkova 2017), sentiment-bearing words from a sentiment lexicon ontology (Wang, Zhang, and Liu 2017), hazardous product reviews (Bhat and Culotta 2017), the detection of spam reviews (Ren, Ji, and Zhang 2014), and the detection of relevant biomedical literature (Li and Liu 2003; Li, Liu, and Ng 2010). To our knowledge, we are the first to suggest PU Learning, particularly using perplexity, for social media relevance filtering.

VAE in NLP: Our use of VAE draws on prior bag-of-words VAE models that were proposed for modelling text and documents (Mnih and Gregor 2014). We use the approach by (Miao and Blunsom 2016) to model the positive class. Here we outline other related VAE approaches for text modelling.

Piecewise constant distribution is proposed as a prior in VAE instead of Gaussian to represent complex latent variables, and this VAE model yields lowest perplexity on the document modelling task (Serban et al. 2017). These generative models are also applied to supervised question answering and dialogue modelling. Recurrent neural network (RNN)-based VAE was proposed to incorporate distributed representations of entire sentences for the tasks of language modelling and imputing missing words (Bowman et al. 2016). Hybrid convolutional-RNN VAE was introduced for the tasks of generating characters and tweets (Semeniuta, Severyn, and Barth 2017). This hybrid VAE encodes long texts better than RNN VAE for character-level generation, and it generates much more diverse tweet samples compared to RNN VAE. Unlike traditional or vanilla deep generative models, VAE produces diverse and well-formed text samples from the prior over latent representations.

Conclusions

We introduced the Perplexity Positive-Unlabelled Learning (PPUL) framework as a means to perform social media relevance filtering, a task well suited for PPUL. We demonstrated how perplexity can be used to identify candidate examples of the negative class, using both a statistical language modelling approach and a Variational Autoencoder. Both methods, when coupled with a logistic regression classifier, generally outperformed strong PU Learning baselines. We demonstrated this on a variety of data sets. In each, PPUL attains state-of-the-art performance and is effective when the amount of positive annotated data is small, working for both event-triggered and topic-triggered social media content.

Acknowledgments

We would like to thank the project's software engineers, Brian Jin and James McHugh, for supporting this research. We also thank the reviewers for their insightful feedback.

References

Barwick, K.; Joseph, M.; Paris, C.; and Wan, S. 2014. Hunters and collectors: seeking social media content for cul-

[13] See comments by BERT contributors on the topic: https://github.com/google-research/bert/issues/35

tural heritage collections. In *VALA2014, 17th Biennial Conference and Exhibition*, 3–6.

Bhat, S., and Culotta, A. 2017. Identifying leading indicators of product recalls from online reviews using positive unlabeled learning and domain adaptation. In *International AAAI Conference on Web and Social Media (ICWSM)*.

Bommannavar, P.; Lin, J.; and Rajaraman, A. 2016. Estimating topical volume in social media streams. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, SAC '16, 1096–1101. New York, NY, USA: ACM.

Bowman, S. R.; Vilnis, L.; Vinyals, O.; Dai, A. M.; Józefowicz, R.; and Bengio, S. 2016. Generating sentences from a continuous space. In Goldberg, Y., and Riezler, S., eds., *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, 10–21. ACL.

Broniatowski, D. A.; Paul, M. J.; and Dredze, M. 2013. National and local influenza surveillance through twitter: An analysis of the 2012-2013 influenza epidemic. *PLOS ONE* 8(12).

Cameron, M., and Sparks, R. 2015. Syndromic Surveillance using Twitter Data. *Emergency Medicine: Open Access* 05(03).

Chen, S. F., and Goodman, J. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Center for Research in Computing Technology, Harvard University.

Comité, F. D.; Denis, F.; Gilleron, R.; and Letouzey, F. 1999. Positive and unlabeled examples help learning. In *Proceedings of the 10th International Conference on Algorithmic Learning Theory*, ALT '99, 219–230. London, UK, UK: Springer-Verlag.

Delort, J.-Y.; Arunasalam, B.; and Paris, C. 2011. Automatic moderation of online discussion sites. *Int. J. Electron. Commerce* 15(3):9–30.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B* 39:1–38.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

Elkan, C., and Noto, K. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, 213–220. New York, NY, USA: ACM.

Erdmann, M.; Ward, E.; Ikeda, K.; Hattori, G.; Ono, C.; and Takishima, Y. 2013. Automatic labeling of training data for collecting tweets for ambiguous tv program titles. In *Social Computing (SocialCom), 2013 International Conference on*, 796–802. IEEE.

Heafield, K.; Pouzyrevsky, I.; Clark, J. H.; and Koehn, P. 2013. Scalable modified kneser-ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 690–696. Sofia, Bulgaria: Association for Computational Linguistics.

Heafield, K. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, 187–197.

Howard, J., and Ruder, S. 2018. Universal language model fine-tuning for text classification. In *ACL*. Association for Computational Linguistics.

Kim, S. M.; Wan, S.; and Paris, C. 2016. Occupational representativeness in twitter. In *Proceedings of the 21st Australasian Document Computing Symposium*, ADCS '16, 57–64. New York, NY, USA: ACM.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. Doha, Qatar: Association for Computational Linguistics.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *CoRR* abs/1312.6114.

Krestel, R.; Werkmeister, T.; Wiradarma, T. P.; and Kasneci, G. 2015. Tweet-recommender: Finding relevant tweets for news articles. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, 53–54. New York, NY, USA: ACM.

Li, X., and Liu, B. 2003. Learning to classify texts using positive and unlabeled data. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, 587–592. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Li, X.-L.; Liu, B.; and Ng, S.-K. 2010. Negative training data can be harmful to text classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 218–228. Cambridge, MA: Association for Computational Linguistics.

Maynard, D., and Funk, A. 2012. Automatic detection of political opinions in tweets. In *Proceedings of the 8th International Conference on The Semantic Web*, ESWC'11, 88–99. Berlin, Heidelberg: Springer-Verlag.

Miao, Y., and Blunsom, P. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 319–328. Austin, Texas: Association for Computational Linguistics.

Miao, Y.; Yu, L.; and Blunsom, P. 2016. Neural variational inference for text processing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, 1727–1736. JMLR.org.

Mnih, A., and Gregor, K. 2014. Neural variational inference and learning in belief networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of

JMLR Workshop and Conference Proceedings, 1791–1799. JMLR.org.

Neubig, G.; Mori, S.; and Mizukami, M. 2013. A framework and tool for collaborative extraction of reliable information. In *Proceedings of the Workshop on Language Processing and Crisis Information 2013*, 26–35. Nagoya, Japan: Asian Federation of Natural Language Processing.

O'Dea, B.; Wan, S.; Batterham, P. J.; Calear, A. L.; Paris, C.; and Christensen, H. 2015. Detecting suicidality on twitter. *Internet Interventions* 2(2):183–188.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Ren, Y.; Ji, D.; and Zhang, H. 2014. Positive unlabeled learning for deceptive reviews detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 488–498. Doha, Qatar: Association for Computational Linguistics.

Rocchio, J. J. 1971. *Relevance Feedback in Information Retrieval*. Englewood, Cliffs, New Jersey: Prentice Hall. 313–323.

Saif, H.; He, Y.; and Alani, H. 2012. Alleviating data sparsity for twitter sentiment analysis. In *2nd Workshop on Making Sense of Microposts (#MSM2012): Big things come in small packages at the 21st International Conference on the-World Wide Web (WWW'12)*, 2–9. CEUR Workshop Proceedings (CEUR-WS.org).

Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J. C.; Smola, A. J.; and Williamson, R. C. 2001. Estimating the support of a high-dimensional distribution. *Neural Comput.* 13(7):1443–1471.

Semeniuta, S.; Severyn, A.; and Barth, E. 2017. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 638–648. Copenhagen, Denmark: Association for Computational Linguistics.

Serban, I. V.; Ororbia, A. G.; Pineau, J.; and Courville, A. 2017. Piecewise latent variables for neural variational text processing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 422–432. Copenhagen, Denmark: Association for Computational Linguistics.

Shen, H.; Bunescu, R.; and Mihalcea, R. 2012. Sense and reference disambiguation in Wikipedia. In *Proceedings of COLING 2012: Posters*, 1111–1120. Mumbai, India: The COLING 2012 Organizing Committee.

Sriphaew, K.; Takamura, H.; and Okumura, M. 2009. Cool blog classification from positive and unlabeled examples. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, 62–73. Berlin, Heidelberg: Springer-Verlag.

Sterckx, L.; Caragea, C.; Demeester, T.; and Develder, C. 2016. Supervised keyphrase extraction as positive unlabeled learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1924–1929. Austin, Texas: Association for Computational Linguistics.

Stowe, K.; Paul, M. J.; Palmer, M.; Palen, L.; and Anderson, K. 2016. Identifying and categorizing disaster-related tweets. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, 1–6. Austin, TX, USA: Association for Computational Linguistics.

Sui, J. 2015. *Understanding and Fighting Bullying with Machine Learning*. Ph.D. Dissertation, Department of Computer Sciences, University of Wisconsin-Madison.

To, H.; Agrawal, S.; Kim, S. H.; and Shahabi, C. 2017. On identifying disaster-related tweets: Matching-based or learning-based? In *Third IEEE International Conference on Multimedia Big Data, BigMM 2017, Laguna Hills, CA, USA, April 19-21, 2017*, 330–337. IEEE Computer Society.

Wan, S., and Paris, C. 2014. Improving government services with social media feedback. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, 27–36. New York, NY, USA: Association for Computing Machinery.

Wan, S.; Paris, C.; and Georgakopoulos, D. 2015. Social Media Data Aggregation and Mining for Internet-Scale Customer Relationship Management. In *Proceedings - 2015 IEEE 16th International Conference on Information Reuse and Integration, IRI 2015*.

Wang, Y.; Zhang, Y.; and Liu, B. 2017. Sentiment lexicon expansion based on neural pu learning, double dictionary lookup, and polarity association. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 564–574. Copenhagen, Denmark: Association for Computational Linguistics.

Xu, J.-M.; Jun, K.-S.; Zhu, X.; and Bellmore, A. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, 656–666. Stroudsburg, PA, USA: Association for Computational Linguistics.

Yang, Y.; Bao, F.; and Nenkova, A. 2017. Detecting (un)important content for single-document news summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 707–712. Valencia, Spain: Association for Computational Linguistics.

Zhang, Z., and Lan, M. 2016. Ecnu at semeval 2016 task 6: Relevant or not? supportive or not? a two-step learning system for automatic detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 451–457. San Diego, California: Association for Computational Linguistics.