


ARTICLE

Learning from noisy out-of-domain corpus using dataless classification

Yiping Jin¹ , Dittaya Wanvarie^{1*} and Phu T. V. Le²

¹Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok 10300, Thailand and ²Knorex Pte. Ltd., 8 Cross St, Singapore 048424, Singapore

*Corresponding author. E-mail: Dittaya.W@chula.ac.th

(Received 8 April 2019; revised 25 May 2020; accepted 25 May 2020; first published online 17 June 2020)

Abstract

In real-world applications, text classification models often suffer from a lack of accurately labelled documents. The available labelled documents may also be out of domain, making the trained model not able to perform well in the target domain. In this work, we mitigate the data problem of text classification using a two-stage approach. First, we mine representative keywords from a noisy out-of-domain data set using statistical methods. We then apply a dataless classification method to learn from the automatically selected keywords and unlabelled in-domain data. The proposed approach outperformed various supervised learning and dataless classification baselines by a large margin. We evaluated different keyword selection methods intrinsically and extrinsically by measuring their impact on the dataless classification accuracy. Last but not least, we conducted an in-depth analysis of the behaviour of the classifier and explained why the proposed dataless classification method outperformed supervised learning counterparts.

Keywords: Text classification; Dataless classification; Noisy labels; Domain adaptation

1. Introduction

Text classification has been extensively studied within the natural language processing (NLP) community, and modern neural networks models achieve promising accuracy of higher than 90% on some well-studied benchmark data sets (Yogatama *et al.* 2017; Howard and Ruder 2018). However, when we apply text classification models to real-world problems, the accuracy is often much lower. There are a few reasons.

First, the text classification models are often trained on a fixed set of training documents. The data labelling process is usually a one-time effort. The static set of training documents does not cover emerging topics or keywords, and there is no guarantee that the model will maintain a decent accuracy after having been deployed for a prolonged period.

Second, off-the-shelf text classification models for a specific domain is often not available, and people settle with a general-purpose classifier or a classifier built for a similar domain. Previous work showed that applying text classification models directly on out-of-domain data will cause a drastic performance drop (Mudinas, Zhang, and Levene 2018).

The third challenge for applying text classification in real-world applications is the lack of accurate training data. Industry projects usually run on a much shorter timeline. Researchers do not have the luxury of time and resources to build a large, high-quality data set (Dahlmeier 2017). They need to settle with either a small data set or use noisy labels obtained from crowd-sourcing or user-generated content such as hashtags (Wang *et al.* 2011).

Last but not least, the Internet contains heterogeneous textual data in the form of static HTML pages or dynamic pages generated using various web frameworks. Different types of web pages such as home pages, forums and product list pages are ubiquitous and have very different characteristics (Nguyen-Hoang *et al.* 2018). This is in contrast to popular benchmark data sets where the data come from a single source. The heterogeneous input may affect the accuracy of the text classification models. However, its impact has not been well studied and quantified.

This work was motivated by our experience building text classifiers for contextual advertising (Jin, Wanvarie, and Le 2017). The goal of contextual advertising is to display ads on only web pages which are related to the ad. We apply text classifiers to categorise all the web pages in the user browsing history. To facilitate the integration with various ad exchanges and publishers, we need to classify the web content into the category taxonomy defined by Interactive Advertising Bureau (IAB), the organisation which develops industry standards for online advertising.^a The taxonomy consists of 23 tier-1 categories and more than 300 tier-2 categories. It covers a broad range of sectors such as automotive, education and travel. The large number of categories, the heterogeneous content on the Internet together, poses a great challenge for advertisers to build and maintain highly accurate text classifiers.

We began building text classifiers for contextual advertising a few years ago by crawling categorised newswire websites such as Reuters^b and Star Online^c and mapping their categories into IAB categories. This saved us a huge effort to create the corpus by labelling web pages manually. However, it suffers some significant drawbacks. Namely, the training data do not resemble the actual user browsing data, and there exists label noise due to human error or imperfect category mapping.

In this work, we propose a method to mitigate the data problem and improve the accuracy of the classifier drastically. We first mine keywords for each category from a noisy labelled training corpus using statistical methods. This is based on the assumption that the label noise of individual documents may offset each other when we calculate the distribution of keywords among a large collection of documents. A robust statistical method shall be able to separate the representative keywords for a category from random words that appear due to noise. Using the keywords, we apply a state-of-the-art dataless text classification model (Li *et al.* 2016) which requires only a handful of seed words for each category and no labelled documents to train. We mitigate the problem of the noisy labels by letting the dataless model figure out the correct label by itself. The dataless paradigm also allows us to learn from unlabelled in-domain documents, which can yield further performance improvement.

Our contributions in this work are threefold. First, we use the automatically mined keywords as the bridge and address the noisy label problem with a dataless learning method. The proposed two-stage approach drastically outperformed various baselines, including a state-of-the-art supervised learning model on data sets for contextual advertising. Second, we conducted a thorough intrinsic and extrinsic evaluation on various keyword extraction methods and their impact on the dataless classifier's accuracy. The proposed method yields both more meaningful keywords and better accuracy for the induced dataless classifier. Lastly, we conducted an in-depth analysis of the working of the classifiers to explain why the proposed method yields superior performance. This provides the basis for further theoretical and empirical studies.

2. Related work

We present three areas of related work which are closely related to this paper, namely text classification with label noise, domain adaptation and dataless classification.

^a<https://www.iab.com/guidelines/iab-quality-assurance-/guidelines-qag-taxonomy/>.

^b<https://www.reuters.com/>.

^c<https://www.thestar.com.my/>.

2.1 Text classification with label noise

There are three main approaches to perform text classification with the presence of label noise: *label noise-robust models*, *data cleansing methods* and *noise-tolerant methods* (Frénay and Verleysen 2014).

Label noise-robust models are the simplest among all approaches. It neither tries to cleanse nor to model the noise. Certain types of classifiers are more robust to the label noise, such as ensembles using bagging (Dietterich 2000). On the other hand, despite being a strong baseline for many classification tasks, support vector machine (SVM) is not robust to label noise (Nettleton, Orriols-Puig, and Fornells 2010). The model relies on a few support vectors close to the decision boundary, and wrongly labelled data can have a large impact on the model's accuracy. Label noise-robust models are relatively effective when the amount of label noise is small.

Data cleansing methods aim first to identify the label noises and then either remove the wrongly labelled data or try to reassign the correct label. Researchers favour data cleansing methods because they can be combined with any standard classification algorithm as an additional preprocessing step. Data cleansing methods are relatively easy to implement. We can either use anomaly detection methods (Sun *et al.* 2007) or model prediction-based filtering with either voting (Brodley *et al.* 1996) or *k*-fold cross-validation (Gamberger, Lavrac, and Groselj 1999).

Last but not least, noise-tolerant methods try to learn a label noise model simultaneously with a classifier. This approach models the label noise explicitly using often a Bayesian prior (Swartz *et al.* 2004; Gerlach and Stamey 2007). In the same spirit, Breve, Zhao, and Quiles (2010) proposed a novel particle walk semi-supervised learning method which is robust to the noise of the label. Their method first converts the data set into a similarity graph and then applies a label propagation algorithm to correct the wrongly labelled instances.

2.2 Domain adaptation

There are two scenarios for domain adaptation, depending on whether there is in-domain labelled data (usually a small amount compared to the original data set) available.

When we have a small amount of in-domain labelled data, transfer learning is the standard approach (Pan and Yang 2010). Transfer learning was popularised through the ImageNet challenges (Krizhevsky, Sutskever, and Hinton 2012). Recently, researchers replicated the success of transfer learning to the field of NLP and achieved new state-of-the-art results for text classification (Howard and Ruder 2018; Peters *et al.* 2018). These approaches first pre-train a language model on a large corpus and then fine-tune the language model using the in-domain unlabelled data. While in-domain labelled data are usually expensive to obtain, unlabelled data such as movie reviews or web pages are often available in abundance. The final step is to train a classifier for the target classification task using the fine-tuned encoder from the previous step. Transfer learning significantly reduced the labelled data needed to train classifiers with decent accuracy. Howard and Ruder (2018) demonstrated that with 100 labelled examples, they could match the performance of training from scratch on 100x more data.

When there is no in-domain labelled data at all, we need to build models that are 'domain-robust' or tap on unsupervised learning methods. Sachan, Zaheer, and Salakhutdinov (2018) investigated various models' reliance on key lexicons by carefully constructing training and testing data sets with not key lexicon overlap. They found out while sophisticated deep learning models can theoretically capture non-local semantic features, they still rely heavily on the presence of keywords in practice. On the lexicon data set, the accuracy of various models dropped on average 10–20%. To reduce this gap, Sachan *et al.* (2018) proposed two methods, namely keyword anonymisation and adaptive word dropout to regularise the model and make it rely less on the keywords. Similarly, Li *et al.* (2018b) performed adversarial training with Gradient Reversal Layer (Ganin *et al.* 2016) to remove category-specific information and to make the model generalise to unseen categories.

Mudinas *et al.* (2018) proposed a novel unsupervised method to bootstrap domain-specific sentiment classifiers. They observed that the positive/negative sentiment words form distinct clusters in the in-domain embedding space. To this end, they trained a simple linear model to classify words into positive or negative sentiment based on their word embedding alone. Then, they used the induced lexicon to assign pseudo-labels to the unlabelled documents. Finally, the pseudo-labelled documents were used to train a supervised long short-term memory model which achieves accuracy comparable to fully supervised approaches.

2.3 Dataless classification

Dataless classification (Chang *et al.* 2008) denotes the family of learning protocols which can induce classifiers without any labelled data (document). Instead, dataless classification algorithms make use of labelled keywords and unlabelled corpus to train the classifier. There are various approaches for dataless classification, such as hand-crafted rules, constraint optimisation, injecting the keywords as priors to the model and semantic representation of the documents and the labels.

Chang *et al.* (2008) made use of *Explicit Semantic Analysis (ESA)* (Gabrilovich *et al.* 2007) to embed both the documents and the labels into a shared semantic space representing concepts inferred from Wikipedia. The classification is performed by calculating the cosine similarity between the document and the label representation. They also considered the impact of dataless classification on domain adaptation. However, differing from this work, they only considered the binary classification between two categories ‘baseball’ and ‘hockey’, and their source and target data set (20NG and Yahoo! Answers data set) are both manually curated and do not contain label noise. Subsequent work on dataless classification extended the ESA approach to both hierarchical classification (Song and Roth 2014; Zheng *et al.* 2016) and cross-lingual classification (Song *et al.* 2016; Song *et al.* 2019).

Druck, Mann, and McCallum (2008) proposed generalised expectation (GE) criteria which induce a classifier by performing constraint optimisation over the distribution of labelled words among documents predicted into each category. GE has been successfully applied on different tasks, such as text categorisation (Druck *et al.* 2008) and language identification in mixed-language documents (King and Abney 2013). Similarly, Charoenphakdee *et al.* (2019) proposed a theoretically grounded risk minimisation framework that directly optimises the area under the receiver operating characteristic curve (area under the curve) of a dataless classification model.

Settles (2011) and Li and Yang (2018) both used multinomial naïve Bayes (MNB) for dataless classification. Settles (2011) extended MNB to allow labels for words by increasing their Dirichlet prior. His method consists of three steps: first to estimate the initial parameters using only the priors; second to apply the induced classifier on unlabelled documents; lastly to re-estimate the model parameters using both labelled and probabilistically-labelled documents. Using an interactive approach to query document and word labels from the user, the system can achieve 90% of state-of-the-art performance after a few minutes of annotation. In contrast, Li and Yang (2018) used the labelled keywords to provide pseudo-labelled documents. They then performed standard semi-supervised learning using expectation maximization algorithm.

Li *et al.* (2016) proposed Seed-Guided Topic Model (STM) for dataless text classification. Different from the standard latent Dirichlet allocation, STM models, two sets of topics: *category topics* and *general topics*. Category topics contains specific words which are representative of a category. General topics are words which frequently occur in a category, but they alone do not indicate the category. For example, if a document contains the keyword ‘mammogram’, it is almost certainly related to cancer. However, it is not the case for keywords like ‘breast’ and ‘prostate’, although they do frequently occur in documents about ‘cancer’. The inference of STM consists of two stages: they first initialise the category word probability and the document category distribution by counting the co-occurrence with seed words belonging to each category.

Then, they apply joint Gibbs sampling to infer all the hidden parameters. STM is demonstrated to drastically outperform various baselines, including GE and a naïve Bayes model similar to Settles (2011). STM has also been extended to perform multi-label classification (Zha and Li 2019) and joint document filtering and classification (Li *et al.* 2018a).

Recently, Meng *et al.* (2018) proposed WESTCLASS, a novel weakly supervised text classification method. It consists of two steps: pre-training and self-training. First, it generates pseudo documents for each category from various sources of supervision, such as labelled keywords or documents. A generative mixture model is used to repeatedly generate a number of terms from a background distribution and the class-specific distribution to form pseudo documents. The pseudo documents are used to *pre-train* a neural model. To adapt to real-world input documents, it performs self-training on unlabelled real documents and automatically adds the most confident predictions to the training set. The method drastically outperformed baselines such as information retrieval with tf-idf, Chang *et al.* (2008) and convolutional neural networks (CNN) trained on pseudo-labelled documents.

Another task closely related to dataless classification is zero-shot text classification (0SHOT-TC) (Yin, Hay, and Roth 2019). Besides allowing no labelled training documents, it requires the model to generalise to *unseen labels*. For example, the classifier is trained on ‘hockey’ and ‘baseball’ category using either a supervised or dataless learning method, and it needs to classify documents belonging to the ‘badminton’ category which occurs only at test time. The main approach to 0SHOT-TC is to calculate the interaction between the document and category embeddings and model it as either a ranking or classification problem.

Li *et al.* (2018) calculated the element-wise difference and element-wise product between the category embedding and each word in the document. The document-level relevance is aggregated using convolutional layers. Nam, Menca, and Fürnkranz (2016) applied a bilinear function $f(x, y)$ in the form of $x^T W y$, where x is the document representation, y is the label representation and W is a matrix with learnable parameters capturing the interaction between the two representations. Pappas and Henderson (2019) proposed generalised input-label embedding (GILE), which extends the bilinear interaction with a more generalised interaction with a non-linear activation function and a controllable parameter capacity. They demonstrated that GILE outperformed the model proposed by Nam *et al.* (2016) drastically for both seen and unseen labels.

We believe 0SHOT-TC is a promising research direction. However, its requirement of generalising to any unseen labels limits the accuracy it can achieve with the current state of research. The state-of-the-art model’s performance on unseen labels is 243–1062% worse than on seen labels based on different evaluation metrics (Pappas and Henderson 2019). In contrast, dataless classification models can often yield performance that is close to a fully supervised model. Besides, it is reasonable to assume that we know the categories in advance before we deploy the classifier. Even if the list of categories is non-static, we can easily retrain the dataless classifier with the new list of category names and keywords. Therefore, we limit the scope of this work to dataless classification and do not consider the zero-shot learning setting.

3. Proposed method

Our proposed method consists of two steps: we first mine keywords for each category from a noisy labelled training corpus and then use a dataless learning algorithm to induce a text classifier with the keywords and unlabelled documents. We provide details of these two steps in Sections 3.1 and 3.2.

Our method was remotely inspired by two well-known principles: the law of large numbers (Hsu and Robbins 1947) and the principle of least effort (Zipf 1949). While individual document labels may contain some random noise, when we collect keyword statistics from a large corpus, we expect the noise to be averaged out, and we can still obtain a set of high-quality keywords

representing each category. On the other hand, by focusing only on keywords, we abstract out the syntactic and contextual information. The induced classifier is less likely to overfit the training corpus and may generalise better to other domains or genres of text. As commented by Settles (2011), learning from keywords is closer to human's learning process and requires much less effort than learning from a large number of labelled documents.

3.1 Mining keywords from (noisy) labelled corpus

The selection of keywords makes a significant impact on the accuracy of the induced dataless classifier (Li *et al.* 2018a). While keyword (or keyphrase) extraction from text has been extensively studied, how the selection of keywords impacts dataless classification was rarely if ever discussed. Previous work used either hand-picked keywords (Druck *et al.* 2008; Settles 2011; Meng *et al.* 2018) or relied on only the category name or category description (Chang *et al.* 2008; Li *et al.* 2016; Li *et al.* 2018b). The problem of extracting keywords from a (noisily) labelled corpus is defined formally as follows.

We have a corpus (D_1, \dots, D_C) , where $D_c = (d_1, \dots, d_k)$ is the set of documents (noisily) labelled as category c . Each document d_i contains a list of terms (w_1, \dots, w_l) . We want to generate a list of representative terms t_1, \dots, t_n from the vocabulary $V = [w_1, \dots, w_N]$ for each category. This is related to the measurement of association in information theory. Therefore, we try to apply pointwise mutual information between keyword w and category c . $pmi(w; c)$ is defined as follows:

$$pmi(w; c) \equiv \log \frac{p(w, c)}{p(w)p(c)} = \log \frac{df(w, c) \sum_{c \in C} df(c)}{df(w)df(c)} \quad (1)$$

where $df(w, c)$ is the number of documents belong to category c and contain word w and $df(w)$ is the number of documents that contain word w . $df(c)$ is the number of documents belonging to category c . Correspondingly, $\sum_{c \in C} df(c)$ is the total number of documents in the corpus. We notice that pmi tends to favour rare words. For example, when a word occurs only once in the corpus, it will have high pmi score in the category where it occurs. This makes the mined keywords unreliable, especially in the presence of the label noise. We therefore introduce $pmi\text{-freq}$ with two modifications: first, we multiple the pmi score with the log-term frequency of word w . Second, we set a threshold of minimum term frequency of 5. The $pmi\text{-freq}$ will be set to zero if the term frequency is below the threshold.

$$pmi\text{-freq}(w; c) = \begin{cases} \log df(w) \log \frac{df(w, c) \sum_{c \in C} df(c)}{df(w)df(c)}, & \text{if } df(w) \geq 5 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

While pmi ensures that there is a strong association between the top keywords and the category, we also want the keywords for different categories to have little or no overlap. We apply maximal marginal relevance (MMR) (Carbonell and Goldstein 1998) to achieve this.

$$mmr \equiv \arg \max_{w_i \in S_{c_m}} [\lambda Sim_1(w_i, c_m) - (1 - \lambda) \max_{w_j \in S_{c_n} \& m \neq n} Sim_2(w_i, w_j)] \quad (3)$$

In Equation (3), the first term measures the similarity between candidate word w_i and category c_m . The second term measures the maximum similarity between candidate word w_i and any seed word from another category c_n . The parameter λ controls the weights of the two terms. A higher λ favours keywords that are strongly associated to category c_m . A smaller λ favours keywords that

occur exclusively in category c_m but not in other categories. We use a default λ of 0.5 and *pmi-freq* as the similarity measure for both Sim_1 and Sim_2 .

We want to study the impact of different label noise rate on the quality of the mined keywords. Since the label noise rate for a corpus is fixed, we synthesise label noise using the following mechanism:

1. choose the label noise rate ϵ to generate noise;
2. calculate the # of docs with corrupted label: $n_{corrupt} = \text{math.floor}(n_{docs} \cdot \epsilon)$; and
3. randomly select $n_{corrupt}$ docs and randomly shuffle their labels;

We use the well-known 20 newsgroups data set (Lang 1995) and vary the percentage of label noise from 0% up to 70%. We manually examine the top 10 keywords to evaluate the quality of the keyword mining algorithm. We count a keyword to be correct if it unambiguously represents the category. For example, while the word ‘Israeli’ represents the category ‘talk.politics.mideast’, the word ‘territories’ does not. Due to the space limit, we only show the results for three randomly selected categories (talk.politics.mideast, rec.autos, rec.sport.baseball), while the other categories follow the same trend.

Among previous work, only Druck *et al.* (2008) proposed an automatic algorithm to mine keywords from oracle-labelled documents based on mutual information (*mi*). Therefore, we compare the three aforementioned methods *pmi*, *pmi-freq*, and *mmr* together with *mi*. Besides, we also show the result of a naïve baseline *freq*, which outputs the most frequent word for each category after stop word removal. *mi* is expressed as

$$mi(w;C) \equiv \sum_{c \in C} p(w, c) \log \frac{p(w, c)}{p(w)p(c)} \propto \sum_{c \in C} df(w, c) \log \frac{df(w, c) \sum_{c \in C} df(c)}{df(w)df(c)} \quad (4)$$

mi is independent from the category since it sums up all the categories. Therefore, Druck *et al.* (2008) first selected the most predictive k features based on *mi* and then assigned the word to the category where it occurs with most often, and other categories that it occurs with at least half as often.

Figure 1 shows the number of correct keywords output by each algorithm for different noise rate ϵ . We can observe that *pmi-freq* and *mmr* almost always generate better keywords than *pmi* except for the automotive category when the label noise rate is relatively low. This is because *pmi* generates specific automotive brand names which are relatively unambiguous. *pmi-freq* and *mmr* remain effective even when the label noise rate is 0.5. They also tend to be more robust against the change of the noise rate, and the generated keywords remain relatively static. On the other hand, *pmi* sometimes generates completely different keywords when the noise rate is increased by 0.1.

Table 1 shows the generated keywords of various algorithms for the category ‘rec.sport.baseball’ with a label noise rate of 0.3. We underline the ambiguous keywords. We can see that almost all keywords generated by *pmi* are person or team names. Of 10 keywords, 6 are ambiguous. In contrast, *pmi-freq* and *mmr* generate specific keywords related to the baseball game. They generate much fewer ambiguous keywords and will likely generalise better. The two baselines *freq* and *mi* both perform poorly. While *freq* tends to generate common words like ‘bad’ and ‘actually’, *mi* tends to generate words that occur frequently in multiple categories (because it sums up the mutual information score for all categories) and therefore have less discriminative power.

The full list of keywords generated by each algorithm at different label noise rate is presented in Appendix A. *Pmi – freq* and *mmr* perform on par with each other. Therefore, we choose to use *pmi-freq* as the final algorithm to mine keywords due to its simplicity.

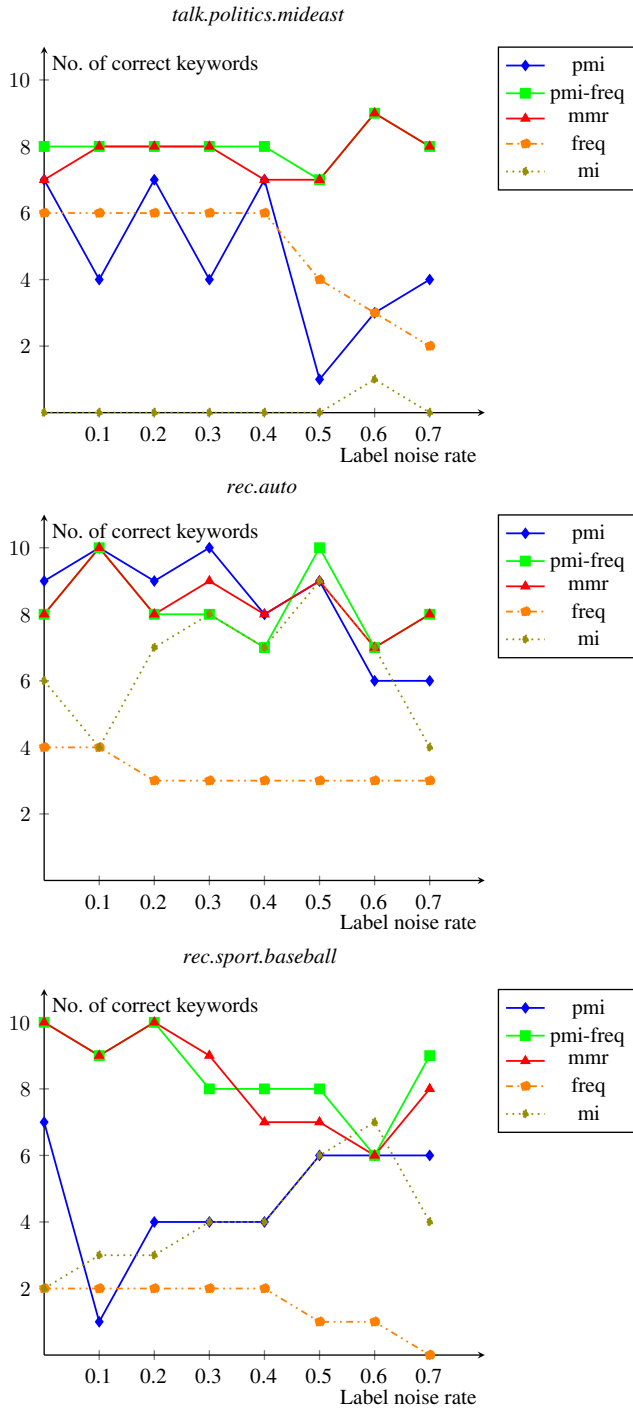


Figure 1. Number of correct keywords generated by each algorithm varying the percentage of label noise.

Table 1. Automatically mined keywords for the category 'rec.sport.baseball' with 0.3 label noise. The ambiguous keywords are underlined

PMI	PMI-FREQ	MMR	FREQ	MI
<u>royals</u>	pitcher	pitcher	baseball	<u>season</u>
<u>hernandez</u>	baseball	braves	<u>games</u>	<u>team</u>
dodgers	braves	pitching	<u>team</u>	baseball
marlins	pitching	pitchers	<u>hit</u>	pitcher
<u>lankford</u>	pitchers	hitter	pitcher	<u>players</u>
braves	hitter	batter	<u>play</u>	braves
<u>fielder</u>	batter	<u>gant</u>	<u>lot</u>	<u>stats</u>
cardinals	<u>gant</u>	inning	<u>league</u>	<u>player</u>
<u>ws</u>	inning	batting	<u>bad</u>	<u>games</u>
<u>winfield</u>	<u>league</u>	jays	<u>actually</u>	hitter

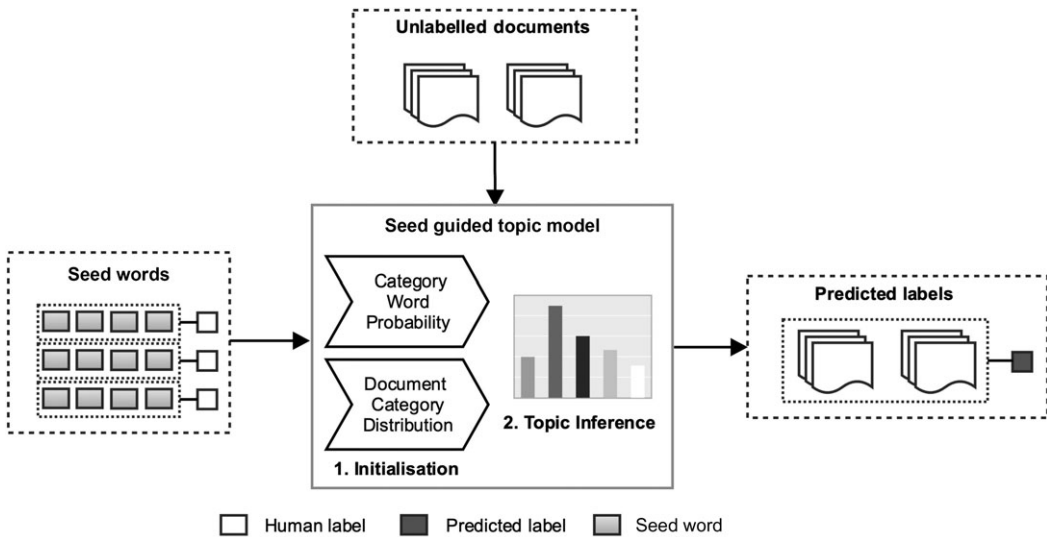


Figure 2. The architecture of the seed-word-guided topic model.

3.2 Training dataless classifiers

We apply the STM (Li *et al.* 2016) to train dataless classifiers. The architecture of STM is depicted in Figure 2. STM takes labelled seed words and unlabelled documents as input. In the first step, the initial document category distribution is estimated from the term frequency of seed words with a Dirichlet smoothing prior. It also calculates the category word probability of unlabelled words based on their co-occurrence with the labelled seed words as follows:

We first calculate the conditional probability $p(w|s)$ using Equation (5), where $df(w,s)$ is the number of the documents containing both unlabelled word w and seed word s . The relevance of

word w to category c is then calculated as the average conditional probability of w with respect to each seed word in \mathbb{S}_c (Equation (6)).

$$p(w|s) = \frac{df(w, s)}{df(s)} \quad (5)$$

$$rel(w, c) = \frac{1}{|\mathbb{S}_c|} \sum_{s \in \mathbb{S}_c} p(w|s) \quad (6)$$

Lastly, the relevance score is normalised by summing over each category c and each word w in the vocabulary in Equations (7) and (8), respectively. The final v_c values are used to initialise the category word probability before the inference process.

$$v(w, c) = \max \left(\frac{rel(w, c)}{\sum_c rel(w, c)} - \frac{1}{c}, 0 \right) \quad (7)$$

$$v_c(w, c) = \frac{v(w, c)}{\sum_w v(w, c)} \quad (8)$$

The model differentiates two types of underlying topics: the category topic and the general topic. General topics capture the global semantic information and are shared by all the documents. A category topic is associated with a single category and captures the relevant keywords of the category. STM model introduces a binary variable $x_{d,i}$ which indicates whether the associated word $w_{d,i}$ is generated from document d 's category topic c_d or from one of the general topics. The parameter inference is carried out using Gibbs Sampling, and the generative process is described below:

1. for each category $c \in \{1 \dots C\}$,
 - a) draw a general-topic distribution $\varphi \sim \text{Dirichlet}(\alpha_0)$ and
 - b) draw a category word distribution $\vartheta \sim \text{Dirichlet}(\beta_0)$;
2. for each general topic $t \in \{1 \dots T\}$,
 - (a) draw a word distribution for the general topic $\phi_t \sim \text{Dirichlet}(\beta_1)$;
3. For each document $d \in \{1 \dots D\}$,
 - (a) generate an initial category distribution η_d ;
 - (b) draw category $c_d \sim \text{Multinomial}(\eta_d)$;
 - (c) draw a general-topic distribution $\theta_d \sim \text{Dirichlet}(\alpha_1 \cdot \varphi_{c_d})$;
 - (d) for each word $i \in \{1 \dots |d|\}$,
 - i. draw $x_{d,i} \sim \text{Bernoulli}(\delta_{w_{d,i}, c_d})$;
 - ii. if $x_{d,i} = 0$: draw word $w_{d,i} \sim \text{Multinomial}(\vartheta_{c_d})$;

if $x_{d,i} = 1$:

 - A. draw general-topic assignment $z_{d,i} \sim \text{Multinomial}(\theta_d)$;
 - B. draw word $w_{d,i} \sim \phi_{z_{d,i}}$.

During the inference/prediction, the model first jointly samples each pair of $x_{d,i}$ and $z_{d,i}$ conditioned on every possible category c . It then estimates the conditional probability distribution $p(c_d = c | \mathbf{z}, \mathbf{x}, \mathbf{c}_{-d}, \mathbf{w})$, where \mathbf{c}_{-d} denotes the collection of documents excluding document d . We observe that STM tends to predict inconsistent labels for the same input. In the work of Li *et al.* (2016), they predict the category as the category sampled from the category probability distribution in the last iteration. Instead, we predict $\text{argmax } p(c_d = c | \mathbf{z}, \mathbf{x}, \mathbf{c}_{-d}, \mathbf{w})$ in the last iteration as the document category.

Table 2. The number of labels assigned to documents in the browsing data set

No. of labels	No. of documents
1	892
2	516
3	84
4	9

The STM model has two main advantages that allow it to achieve high accuracy for dataless classification. First, the model explicitly calculates the correlation of unlabelled words to the seed words and uses it to initialise the category word probability. This makes the inference process much easier compared to using a randomly initialised probability distribution. Second, by separating the topics into general topics and category topics, the model can focus on only the reliable signals and ‘skim-through’ the rest of the document by assigning it to a general topic.

We apply the STM model with the keywords mined using the algorithm described in Section 3.1 and a large unlabelled corpus to train the final classifier.

4. Experiments

4.1 Data sets

We use three data sets to carry out experiments in this work. One is a legacy large labelled data set crawled from newswire sites. We refer to this data set as *news-crawl data set*. The label was obtained by mapping the news categories to IAB categories. Therefore, we expect the presence of label noise in the data set. We also crawled another evaluation data set with roughly one hundred documents per category following similar methodology. We refer to it as *news-crawl-v2 data set*. These two data sets differ in two ways. First, *news-crawl data set* was collected before April 2015, and *news-crawl-v2 data set* was collected during May 2019. Second, they are crawled from different websites. These differences allow us to study the behaviour of the models when applied to a slightly different domain. The details of constructing the data set as well as the websites where the two data sets were collected are presented in Appendix B.

The third data set is a small manually labelled evaluation data set. We crawled the documents from URLs in the real-time-bidding (RTB) requests we logged. The RTB traffic contains URLs in the user browsing history where there is an opportunity for us to display ads. It contains heterogeneous web pages, such as forums, blogs and even social network sites. This data set is more realistic to our application domain. We refer to this data set as *browsing data set*. All data sets contain the same 22 categories (all IAB tier-1 categories^d except for ‘News’ because ‘News’ can cover any topic). We release the evaluation data sets publicly for researchers to reproduce our results and to facilitate future research in contextual text classification.^e

For the browsing data set, we found out during the annotation that some documents may belong to multiple categories. Therefore, we did not limit to one category per document but labelled all the correct categories. Table 2 summarises the number of labels assigned to documents. Sixty per cent of the documents were assigned only one label, and ninety-four per cent of the documents have two or fewer labels. Multi-label classification is beyond the scope of this work. We are only interested in predicting one of the correct labels which have been annotated.

^d<https://www.iab.com/guidelines/iab-quality-assurance-/guidelines-qag-taxonomy/>.

^e<https://github.com/YipingNUS/nle-supplementary-dataset>.

Table 3. Statistics of data sets. The categories are sorted by the number of documents in the news-crawl corpus

Category label	News-crawl	News-crawl-v2	Browsing
Business	44,343	100	50
Society	25,460	89	71
Technology and computing	16,466	100	178
Health and fitness	16,171	100	132
Law, government and politics	14,374	97	44
Science	11,863	100	96
Sports	11,055	100	92
Art and entertainment	10,746	100	207
Education	8321	100	80
Personal finance	5693	80	56
Automotive	5522	91	109
Food and drinks	4408	100	173
Family and parenting	4204	118	44
Style and fashion	4191	100	62
Travel	3995	100	135
Hobby and interest	3710	100	117
Pets	3246	100	22
Religion and spirituality	2936	95	57
Home and garden	2427	100	66
Real estate	2056	100	86
Careers	1685	65	49
Shopping	1611	92	152
Total	204,483	2127	1501

The number of documents in the three corpora is shown in Table 3. We can see that the number of documents for each category is imbalanced, especially for the news-crawl data set. We did not downsample the majority categories but kept all the documents that we crawled. Another observation is that the categories with the most number of documents in the news-crawl data set and browsing data set are very different. While the news-crawl data set contains many documents related to business or politics, in the user browsing data set, there are more documents related to entertainment, food&drinks and shopping. Besides, there is also a difference in the document length in the data sets. The average document lengths for the news-crawl data set and news-crawl-v2 data set are 503 and 1470 words, while in the user browsing data set, it is 350 words. This evidence suggests a potential mismatch between the training data and the real-world data where the model is applied.

4.2 Experimental setup

We split news-crawl data set randomly into a 0.9/0.1 training and testing set. The fixed test set is only used for evaluation and not included during training and keyword mining. News-crawl-v2 and browsing data set are reserved for evaluation only.

4.2.1 Parameter setting

For the first step of mining keywords from the labelled data set, we select the top 15 unigram keywords for each category based on the *pmi-freq* score. All the documents have been lowercased, and we remove keywords that are less than four characters, contain numbers or contained in NLTK stopword list.^f The remaining keywords and the training data (labels removed) are used to train the STM model. We use the parameters recommended in Li *et al.* (2016) with the total number of topics T being three times the number of categories. We set $\alpha_0 = 50/T$, $\beta_0 = \beta_1 = 0.01$, $\alpha_1 = 100$ and $\rho = 0.8$. We stop the inference after the 5th iteration.

4.2.2 Methods in comparison

We compare our proposed method against various supervised learning and dataless classification baselines and a recent transfer learning-based state-of-the-art model. All the models use word unigrams as features and weight using term frequency when applicable.

Supervised learning baselines:

- **MNB:** MNB model is a competitive baseline for text classification tasks (Wang and Manning 2012). We train a supervised MNB model with Laplace smoothing ($\alpha = 1$). We use the implementation in scikit-learn.^g
- **SVM:** SVM is a versatile model used widely across different domains. It is also one of the most commonly used machine learning models in the industry. We train a linear SVM classifier using stochastic gradient descent with the default parameter settings in scikit-learn ($\alpha = 1e - 4$). We use term frequency as the weighting scheme.
- **K-nearest neighbours (KNN):** We train a KNN model with a relatively small $k = 3$. In practice, KNN's prediction time is at least two magnitudes slower than the other models, making it not applicable for production usage. We show the results of this model for comparison purpose only.
- **Universal language model fine-tuning (ULMFiT)^h:** ULMFiT (Howard and Ruder 2018) is a recent model applying transfer learning to induce a text classifier from a language model. It reported state-of-the-art performance on various topic and sentiment classification benchmarks. We use the implementation of ULMFiT in fastai libraryⁱ and apply the optimisation tricks such as discriminative fine-tuning and gradual unfreezing as proposed by Howard and Ruder (2018). We use the default parameters and fine-tune the classification model for 15 iterations using our training data.

Dataless classification baselines:

- **GE:** GE (Druck *et al.* 2008) is a dataless classification method using user-labelled keywords to constrain the training of a discriminative model. We use the same user labelled keywords as our proposed method. The GE implementation is from the Mallet library.^j

^f<https://www.nltk.org/data.html>.

^g<https://scikit-learn.org>.

^hAlthough ULMFiT is pretrained on unlabelled corpus, it still requires labelled documents in the fine-tuning step. Therefore, we count it as supervised method.

ⁱ<https://github.com/fastai/fastai>.

^j<http://mallet.cs.umass.edu/>.

- **MNB with priors (MNB/Priors):** MNB/Priors (Settles 2011) is another dataless classification baseline which increases priors for labelled keywords and learns from an unlabelled corpus using EM algorithm. We use the open-source MNB/Priors implementation provided by the author.^k
- **Doc2vec:** Doc2vec (Le and Mikolov 2014) learns distributed embedding representation of documents. We concatenate the keywords for each category to form a ‘document’ and infer its document vector as the ‘category vector’. When predicting the category of a document, we simply take the category of the nearest category vector. We use the doc2vec implementation in gensim^l with default parameters. We set the embedding size to 100 and train the model for 10 epochs.
- **WESTCLASS:** WESTCLASS (Meng *et al.* 2018) is a weakly supervised neural model for text classification. Although WESTCLASS can take various types of supervision, we limit to the keywords for a fair comparison with other dataless classification methods. We use the implementation by the original authors.^m We generate 500 pseudo documents per category for pre-training,ⁿ and the entire unlabelled training corpus for self-training. We also use the CNN architecture as recommended in the paper.

4.2.3 Performance metrics

For the news-crawl data set, we report the accuracy and Macro- F_1 scores. Because the categories are highly imbalanced, Macro- F_1 is more meaningful than Micro- F_1 to indicate the average performance across different categories.

For the browsing data set, because some documents contain multiple labels, we cannot apply standard multi-class classification metrics directly. Therefore, we use accuracy⁺ and ma F_1 for multi-label classification following Nam *et al.* (2017).

Accuracy⁺ is defined as the total correct predictions divided by the total predictions. Since all the models are multi-class classification models and predict only one label, we count the prediction to be correct if the predicted label is one of the labels that has been annotated by the human annotator. ma F_1 for multi-label classification is defined as

$$\text{ma}F_1 = \frac{1}{L} \sum_{j=1}^L \frac{2tp_j}{2tp_j + fp_j + fn_j} \quad (9)$$

where L is the number of categories and tp_j , fp_j , and fn_j denote the number of true-positive, false-positive and false-negative of category j . We note that to obtain a perfect ma F_1 score of 1, the model needs to predict all the correct label(s) for each document. Since all the models in comparison predict only one label for each document, the ma F_1 score is strictly lower than 1, but the comparison is still fair nevertheless.

4.3 Results and discussion

4.3.1 Mined keywords from labelled corpus

Table 4 shows the generated keywords for each category, which will be used by all dataless classification models.

^k<https://github.com/burrsettles/dualist>.

^l<https://radimrehurek.com/gensim/models/doc2vec.html>.

^m<https://github.com/yumeng5/WeSTClass>.

ⁿMeng *et al.* (2018) demonstrated that generating more than 500 documents per category will not yield any performance improvement.

Table 4. Generated keywords using pmi-freq

Category	Generated keywords
Business	<i>aircraft railways ridership airframe airbus commuters aviation harvesting railroads roofing marketers boeings</i>
Society	<i>skout matchcom okcupid friendships transgender samesex lesbian marriages flirt lgbt dating lesbians heterosexual</i>
Technology and computing	<i>android scan apps firmware samsung os leftright device smartphones keyboard snapdragon 64bit usb smartphone</i>
Health and fitness	<i>symptoms inflammation medications disease vitamin disorders diabetes diet chronic diagnosis nutrition infections</i>
Law, government and politics	<i>immigration passport uscis embassy attorney lawyers consular citizenship consulate lawyer legal citizens immigrants</i>
Science	<i>horoscopoe astrology atoms earths jupiter planets nasa molecules electrons telescope particles forecast orbit</i>
Sports	<i>olympics medal league semifinal finals midfielder freestyle championship semifinals football stadium athletes</i>
Art and entertainment	<i>bollywood actress actor films film song album singer actors songs lyrics comedy costar drama movie hollywood</i>
Education	<i>colleges universities students exam academic undergraduate admissions faculty examination cbse campus education</i>
Personal finance	<i>stocks investors securities nasdaq equity dividend investor bse earnings trading nse volatility bluechips intraday</i>
Automotive	<i>torque tires honda brakes wheels v8 exhaust transmission chevrolet steering engine cylinder dealership mileage sedan</i>
Food and drinks	<i>recipe sauce bake preheat recipes flour butter delicious flavor ingredients vanilla baking cheese stir garlic</i>
Family and parenting	<i>babys babycenter pregnancy babies trimester baby uterus pregnant breastfeeding placenta midwife newborn</i>
Style & Fashion	<i>calories tattoo weightloss fat waistline dieting menswear acne sneaker carbs cardio dresses slimming moisturising</i>
Travel	<i>kayak booking rentals airline hotels attractions beaches resorts reservation reservations couchsurfing hotel</i>
Hobby & Interest	<i>minecraft armor gameplay quests puzzle ingame multiplayer rpg enemies crossword weapons pokemon monsters</i>
Pets	<i>puppies vet puppy breeds dogs veterinarian breed dog pups breeders kennel pet terrier cats canine</i>
Religion & Spirituality	<i>christians christ jesus bible religious worship islam christianity quran muslims church prayer scriptures muslim</i>
Home & Garden	<i>diy wood soil gardeners cabinets backsplash mulch planting compost plants fertiliser decor watering screws potting</i>
Real-estate	<i>furnished rent condo bedrooms rental sqft apartments apartment bedroom spacious trulia renovated vrbo rentals</i>
Careers	<i>vacancies recruitment candidates interviewer resume qualification employers employer freshers vacancy interviewers</i>
Shopping	<i>coupons coupon pricepony discount scoopon cashback freebies storewide</i>

Table 5. Performance of various models on three data sets: news-crawl test set, news-crawl-v2 data set and on browsing data set. The best results on each data set are in boldface

Model	News-crawl test set		News-crawl-v2		Browsing data set	
	Accuracy	Macro- F_1	Accuracy	Macro- F_1	Accuracy ⁺	ma F_1
Random	0.045		0.045		0.067	
Most frequent	0.217		0.055		0.146	
MNB	0.817	0.766	0.524	0.466	0.660	0.504
SVM	0.850	0.811	0.489	0.470	0.471	0.381
KNN	0.751	0.679	0.189	0.159	0.166	0.103
ULMFiT	0.922	0.892	0.541	0.496	0.564	0.431
GE	0.510	0.483	0.596	0.587	0.777	0.617
MNB/Priors	0.533	0.411	0.439	0.366	0.631	0.493
Doc2vec	0.391	0.383	0.480	0.461	0.557	0.424
WeSTCLASS	0.187	0.163	0.190	0.158	0.177	0.121
STM	0.544	0.527	0.623	0.607	0.794	0.625

4.3.2 Text classification performance

Table 5 shows the performance of various models on the three data sets. All the supervised learning models are trained on the full news-crawl training set with class labels. All the dataless classification models are trained using the same list of keywords in Section 4.3.1. They also access the full news-crawl training set but without the class labels.

All the supervised models performed reasonably well on the news-crawl test set. However, their performance degraded drastically on the other two data sets. This shows that while supervised learning methods can learn important features from the training data and predict accurately on similar documents, there is no guarantee that the model will perform well when the input document looks very different, although they are about the same topics. The recent state-of-the-art ULMFiT model outperformed the other baselines by a large margin on news-crawl test set and achieved a high accuracy of 0.922. Its performance on the other data sets is still competitive among the supervised baselines but lags behind the best dataless classification models.

While MNB's performance on the news-crawl test set lagged behind SVM, its performance on the other two data sets was superior, suggesting that it generalises better to a new domain different from the training data. This is consistent with the finding of Sachan *et al.* (2018), where a discriminative Logistic Regression model suffered more than a Naïve Bayes model when applying on a corpus without important lexicon overlap. The KNN model obtained a slightly inferior yet still reasonable performance on the news-crawl test set. However, it failed on the other two data sets. Suggesting the differences between the data sets are large, and a similarity-based classification algorithm will not work.

It is interesting to observe that while dataless classification models lagged behind all the supervised learning models on the news-crawl test set, they yielded competitive performance on the other two data sets. This confirms our intuition that by abstracting the semantics using keywords, we can obtain better transferability.

STM achieved the best performance on all three data sets compared to other dataless baselines. GE is the second best model, and its performance is consistently 1–3% lower than STM. Despite the two models have completely different architectures, they both explicitly exploit the

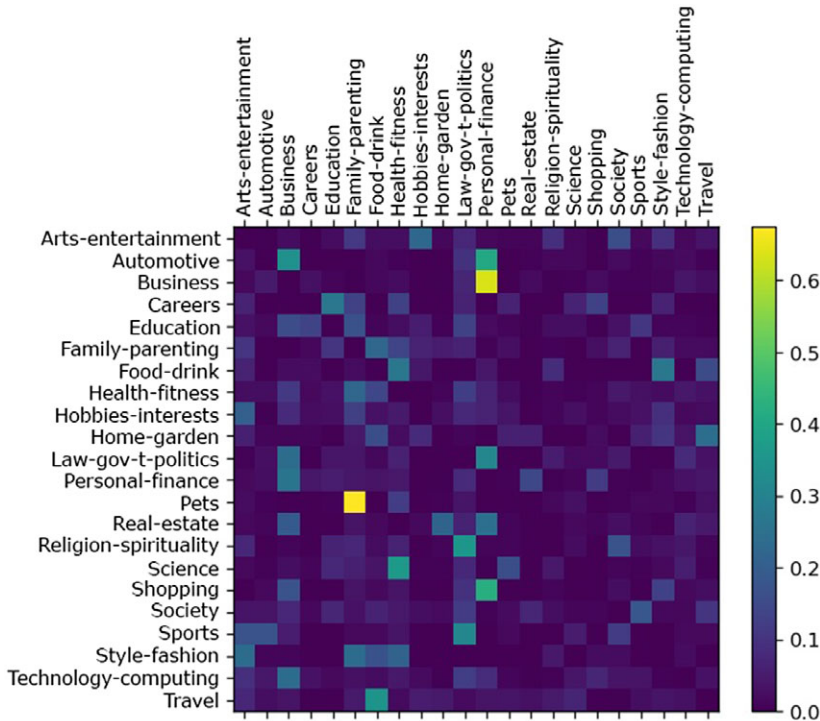


Figure 3. The confusion matrix of STM model prediction on the news-crawl test set. The diagonal entries (correct predictions) have been removed to surface the misclassifications.

word-occurrence information, suggesting that it is an important strategy to bootstrap knowledge in a dataless learning process.

The doc2vec baseline has a mediocre performance on all data sets and is consistently 15–20% lower than STM. WESTCLASS’s performance was surprisingly very low. Meng *et al.* (2018) demonstrated that the model performed well on binary sentiment classification and topic classification with a small number of categories (four or five). However, our task has 22 categories. The core assumption of WESTCLASS that the keywords and documents of each category lie in disjoint clusters in a low-dimensional space may not hold when the number of categories get larger. This is partially validated by the poor performance of doc2vec, where the average of the keyword embeddings is used to represent the category. After pretraining with pseudo-labelled documents, WESTCLASS has a poor macro- F_1 score of 0.104, suggesting the poor quality of the pseudo documents. The self-training does improve macro- F_1 by nearly 6%, but WESTCLASS’s performance remains very poor compared to other baselines.

Two questions arose naturally when we were analysing the result:

1. Why STM performs well on different data sets but not so well on the test set which is most similar to the data which it is trained on?
2. What caused ULMFiT’s performance to degrade drastically when applied on another domain?

To answer the first question, we plot the confusion matrix of STM model on news-crawl test set in Figure 3. We can see that the misclassifications are not random. While we anticipate misclassifications among closely related categories such as ‘business’ and ‘personal-finance’, some other

Table 6. Top five categories the STM model predicts for documents belonging to pets category

Category	No. of predictions
Pets	194
Family and parenting	89
Health and fitness	16
Law, government and politics	5
Travel	4

Table 7. Example documents with label ‘pets’ and are classified as ‘family & parenting’

#	Text
1	poll Do you think it's important for children to have pets? No, pets only make messes Yes, it teaches them responsibility. Share your vote on facebook so your friends can take this poll
2	Babies versus pets in viral advertising posted which do you prefer? Pets or babies? They're everywhere in social media pulling views sparking massive followings rising to the top of every hit list it's a massive love fest huh? what's going on? have we gone cute crazy? why do these characters work so well? . . .
3	'The dog is (by which you mean, 'I want a divorce!') . . . The dog is bored is my husband projecting? transferring? planning on taking the dog for a romantic tropical vacation? Am I right? Am i crazy? You decide. Relationships are full of mystery and are open to interpretation, wild speculation and deep neurosis . . .

cases are worth investigating, such as misclassifying a large proportion of ‘pets’ documents to ‘family-parenting’.

To this end, we did further analysis on the documents belonging to the ‘pets’ category that are misclassified as other categories. Table 6 shows the top five categories of STM model predicted for documents belonging to the ‘pets’ category. Our first impression is that these categories are somehow related to ‘pets’, such as pets are part of the family and important especially to children. Some articles may also talk about veterinary medicine or pet-related diseases, thus making it related to ‘health & fitness’ category.

We inspected the documents which are ‘misclassified’ as ‘family & parenting’ and found that almost all of them are related to children and pets or pets in a family/relationship. Table 7 shows some example snippets. These documents naturally belong to both categories, but only one label was assigned in the news-crawl data set. This explains why STM’s performance on the news-crawl test set is poor, while it performs well on the browsing data set, where all correct categories are labelled.

To understand why ULMFiT’s performance dropped significantly when applied on a different domain, we want to understand what features the model learns and whether these features can be easily transferable across domains. Therefore, we used LIME^o (Ribeiro, Singh, and Guestrin 2016), a model-agnostic interpretation technique to explain the predictions of ULMFiT on sample text drawn from different data sets. LIME perturbs the input instance X slightly and probes the classifier for prediction. It then fits a linear interpretation model that approximates the classifier locally in the vicinity of X . LIME can output a list of features contributing to predicting each category with their feature importance.

^o<https://github.com/marcotcr/lime>.

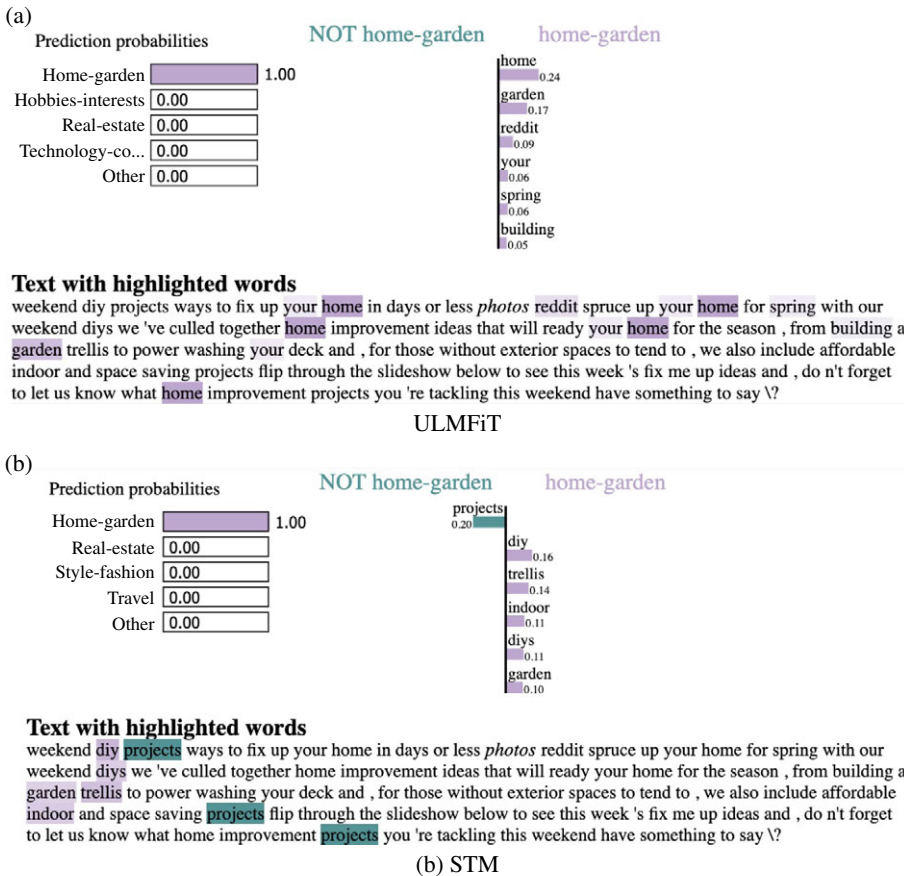


Figure 4. LIME explanations on a sample document from news-crawl data set.

Figure 4 shows LIME’s explanations for both ULMFiT and STM on a sample document from news-crawl data set.^P While both models predict close to 1.0 probability for the correct category ‘home-garden’, the interpretation for STM is obviously more plausible. In contrast, Figure 5 depicts an example from browsing data set where STM predicts the correct label with high confidence but ULMFiT predicts the wrong label. In general, we found that ULMFiT tends to focus on more ‘fuzzy’ features. This may due to the nature of deep learning models which capture complex interactions of non-local features. While these features helped ULMFiT to achieve a very high accuracy on a random-split test set, they may not remain reliable when the input data differ significantly from the training data.

4.3.3 Impact of keyword selection strategy

In Section 3.1, we manually evaluated the quality of the mined keywords using different algorithms. In this section, we try to answer the question how much different keyword selection strategies affect the accuracy of the induced dataless classifier and whether our proposed keyword selection method improves the final accuracy.

^PThe document has been truncated due to space limit but sufficient information is left for the models to make the correct prediction.

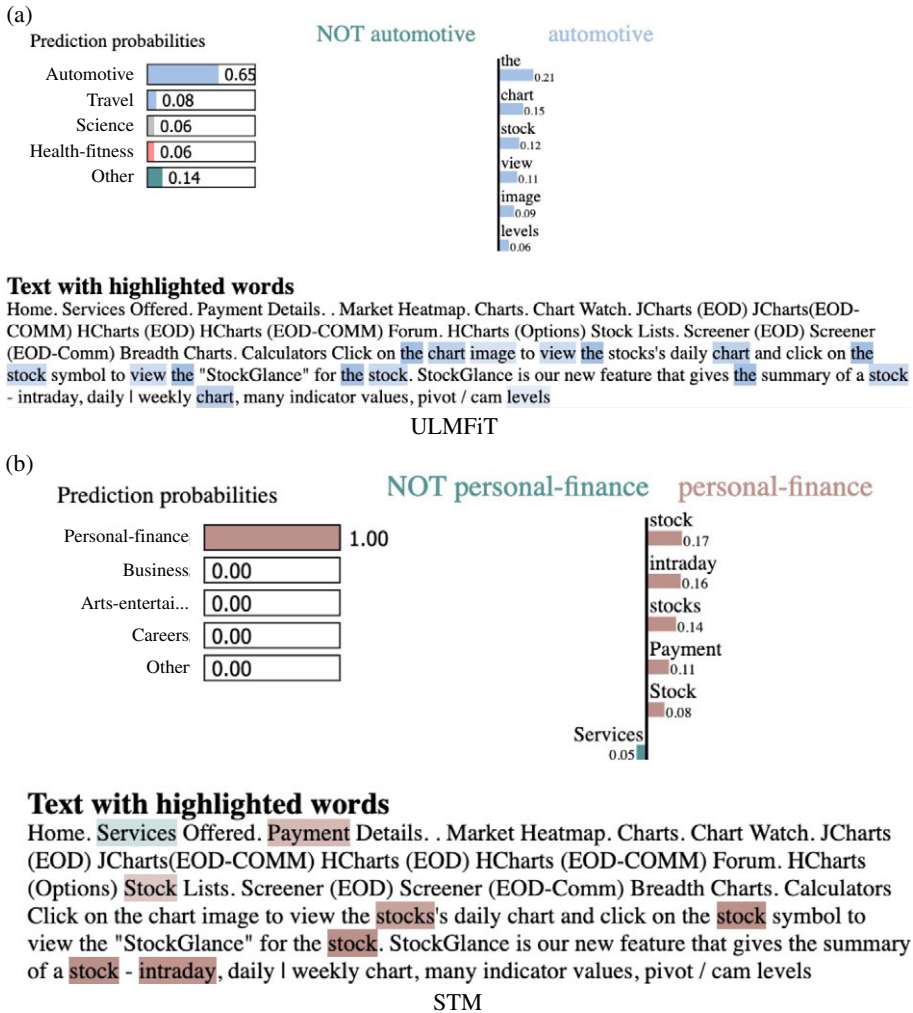


Figure 5. LIME explanations on a sample document from browsing data set.

To this end, we trained STM with different set of keywords and evaluated them on the same evaluation data sets. The keyword selection strategies we compared with are S_{label} , which uses only the words occurring in the category name. It is one of the systems used in Li *et al.* (2016). We also compare with S_{freq} and S_{mi} , which use frequency-based and mutual information-based keyword selection mentioned in Section 3.1. For S_{freq} and S_{mi} , we generate 15 top keywords for each segment, making the number of keywords equal to $S_{pmi-freq}$. We publish the keywords selected using each method to facilitate replication of our results.⁹

Table 8 shows the result. First, using the seed words occurring in the category name alone resulted in the poorest performance, indicating that the category name does not provide sufficient supervision to train an accurate classifier. Some category names are covering terms which might not frequently occur in the text, such as ‘technology & computing’, where people might talk much more about ‘mobile phones’ and ‘laptops’ than ‘computing’. The baseline seed word selection methods S_{freq} and S_{mi} improved from S_{label} . However, their performance still far lagged behind

⁹<https://github.com/YipingNUS/nle-supplementary-dataset>.

Table 8. Performance of STM using different set of keywords on three data sets: news-crawl test set, news-crawl-v2 data set and browsing data set

Model	News-crawl test set		News-crawl-v2		Browsing data set	
	Accuracy	Macro-F ₁	Accuracy	Macro-F ₁	Accuracy ⁺	maF ₁
STM + $S_{pmi-freq}$	0.544	0.527	0.623	0.607	0.794	0.625
STM + S_{label}	0.270	0.243	0.332	0.259	0.405	0.340
STM + S_{freq}	0.284	0.257	0.425	0.359	0.500	0.358
STM + S_{mi}	0.301	0.265	0.434	0.344	0.565	0.385

Table 9. Sample generated keywords using *mi* and the P/R/F₁ on the news-crawl-v2 data set

Category	Generated keywords	P/R/F ₁
Family and parenting	<i>pregnancy babys breastfeeding uterus babycenter vaginal fetus trimester cervix pediatrician</i>	0.76/0.91/0.83
Real estate	<i>clicked movingcom realtorcom hdb blk eunos yishun kio lebar foreclosures</i>	0/0/0

the proposed $S_{pmi-freq}$, demonstrating the importance of the seed word selection method on the final accuracy of the dataless classifier.

Interestingly, we observed that *mi* did generate good keywords for some categories but failed for some other categories. We show two sample categories in Table 9 with their corresponding keywords selected by *mi* and their P/R/F₁ score on the news-crawl-v2 data set. Most keywords *mi* selected for the category ‘real estate’ turned out to be location names in Singapore. This might be due to the bias in the data collection. As a result, the category did not generalise at all and had a zero F₁ score on the news-crawl-v2 data set. On the other hand, *mi* generates good keywords for the category ‘family & parenting’, and the F₁ score is also high. It demonstrates that *mi* is to some extent effective to detect meaningful keywords. However, it is not robust enough to guarantee good-quality keywords for each category.

Putting together the result in this section and Section 4.3.2, we want to highlight that the selection of seed words has at least as much impact on the dataless classification accuracy as the selection of different algorithms. However, it has not received due attention from the research community.

4.4 Experiment on sentiment classification data sets

To demonstrate the effectiveness of the proposed method on other tasks and data sets, we conduct an experiment on a pair of publicly available sentiment classification data sets. We train the supervised and dataless classifiers on the IMDB data set (Maas *et al.* 2011) and evaluate the model on both IMDB and Yelp test set (Zhang, Zhao, and LeCun 2015). Both data sets consist of evenly distributed binary polarity classes. IMDB data set contains 25,000 training documents, 25,000 testing documents and an additional 50,000 unlabelled documents. Yelp test set contains a total of 38,000 documents. Compared to the previous data sets for contextual advertising, these data sets are easier for three reasons: they contain only two classes; the classes are balanced and the document labels are free from noise.

We use IMDB training set to train all the supervised classifiers.[†] We also use the training set to automatically mine 30 keywords for each class, which are used by all dataless classifiers. The

[†]We use the combination of training set and unlabelled set to fine-tune the language model for ULMFiT.

Table 10. Generated keywords using pmi-freq from IMDB training set

Category	Generated keywords
Negative	<i>worst waste awful poorly pointless terrible worse horrible lame stupid crap laughable redeeming unfunny wasted bad boring badly pathetic mess ridiculous dull atrocious incoherent lousy poor supposed garbage sucks unwatchable</i>
Positive	<i>excellent wonderful superb wonderfully beautifully amazing perfect touching captures fantastic flawless delightful favourite terrific refreshing superbly perfection outstanding gem underrated breathtaking brilliant loved finest excellently highly favourites friendship brilliantly matthau</i>

Table 11. Accuracy of various models on IMDB and Yelp test set. The best results on each data set are in boldface

Model	IMDB		Yelp	
	Original	Curated	Original	Curated
MNB	0.814	–	0.745	–
SVM	0.840	–	0.773	–
KNN	0.611	–	0.539	–
ULMFIT	0.944	–	0.856	–
GE	0.813	0.810	0.767	0.781
MNB/Priors	0.802	0.802	0.777	0.774
Doc2vec	0.645	0.613	0.661	0.636
WeSTCLASS	0.698	0.696	0.653	0.684
STM	0.792	0.792	0.713	0.705

list of keywords are shown in Table 10. We then use the combination of the training set and the unlabelled set to train the dataless classifiers. Table 11 shows the results of all the competing models. Since the data sets contain only two evenly distributed classes, we report only the accuracy score.

We can make a few observations from the result. First, some dataless classification models perform on par with simple supervised learning baselines. This is encouraging because the supervised models are trained using 25,000 labelled documents, while the dataless classification models use only 30 automatically mined keywords per category. Second, ULMFiT performs the best on both data sets. This is probably because the training data are free from noise, and the two data sets are relatively similar (both are user reviews).

While dataless classification models do not perform as well as a state-of-the-art supervised model on sentiment classification data sets, they do demonstrate better robustness when applied to a different domain. On average, supervised models' accuracy dropped 7.5% when applied to Yelp data set compared to IMDB test data set. On the other hand, dataless classification models' accuracy dropped only 3.6%.

We believe the performance of dataless classification models versus supervised models is related to the bias-variance tradeoff. Dataless classification models have high bias resulting from the labelled keywords, but low variance, and can generalise better to samples that look different. Supervised models, especially deep learning models, have much lower bias but high variance. It gives us the hint that dataless classification models might perform better than supervised learning models when the document labels contain a lot of noise or the training and testing samples look very different (high covariate shift).

In this set of experiments, STM does not perform as well as GE and MNB/Priors. STM applies topic modelling to capture latent topics in the background corpus. It might be more suitable for topic classification rather than sentiment classification. In the previous topic classification experiments on contextual advertising data sets, despite the input source changes, the underlying topics remain similar and therefore the inferred topics are useful across different data sets. However, the IMDB data set consists of movie reviews, and the Yelp data set consists of reviews for points of interest such as restaurants and hotels. The topics in the two data sets are completely different, suggesting that the latent topics STM inferred from the IMDB data set are not transferable to the Yelp data set. This explains why STM's accuracy dropped close to 8% on the Yelp data set.

The underlined keywords in Table 10 are considered low quality. They consist of movie or actor names ('redeeming' and 'matthau'), movie-specific words ('captures' and 'unwatchable') and general words ('supposed' and 'friendship'). Mentioning a movie or an actor may be a signal whether the review is positive or negative, but they are mostly irrelevant in another domain. Therefore, we want to study the impact of curating the automatically mined keywords on the accuracy of data-less classification models. In Table 11, we show the result with both the original list of keywords in Table 10 and the curated keywords after removing all the underlined domain-specific and noisy keywords. As we expected, there is no improvement on the IMDB test set after curating the keywords. In the Yelp data set, GE and WESTCLASS's accuracy improved while the other models' accuracy either remained the same or decreased. No conclusion can be drawn but we believe in a real-world application setting, it is worthwhile curating the keywords, especially when the original list of keywords is noisy.

5. Domain adaptation performance

Our models are trained using the news-crawl data set and aim to be applied to the data similar to the user browsing data set. Since there is a clear sign of mismatch between the two domains, we are interested in studying how can unlabelled in-domain user browsing data help to train more accurate models. While labelling a large amount of in-domain data can be prohibitively expensive, unlabelled in-domain data are often available in abundance.

To this end, we crawled an additional 280 thousand URLs from the user browsing history in the RTB log and created the *browsing-unsup* data set (without overlap with the browsing data set). Applying the STM model on the unlabelled data set is straightforward. We used the same set of keywords and the new unlabelled in-domain data set to train the new model. For ULMFiT, we applied two different strategies, namely *mix-domain* and *cross-domain*. In the mix-domain setting, we used the (unlabelled) in-domain data to fine-tune the language model and used the original news-crawl data set to train the classifier. In the cross-domain setting, we used self-training similar to Meng *et al.* (2018). We first applied the previous ULMFiT model trained on the news-crawl data set to predict labels on the browsing-unsup data set. We then used the pseudo-labelled documents to train a new ULMFiT classifier.⁵ In this way, both the data to fine-tune the language model and the data to train the final classifier are from the in-domain data. We compare the accuracy of the classifiers being trained on the news-crawl and the in-domain browsing-unsup data in Table 12 (with the relative percentage of change in the bracket).

We can clearly observe that the STM model benefited from unlabelled in-domain data. This is because while the representative keywords may occur in both data sets, the context they appear may differ. By tapping on the unlabelled in-domain data, the model can capture features which are useful in the target domain.

As we expected, the ULMFiT mix-domain model did not improve the performance. This is possibly because we had to use different sources of data to fine-tune the language model and to train the final classifier. The fine-tuned language model might have been 'unlearned' when training

⁵We used only documents with a label probability higher than 0.8 to ensure the labels are relatively clean while we have at least 1000 documents for each category.

Table 12. Impact of unlabelled in-domain data on the model performance. The best results on each data set are in boldface

Model	News-crawl test set		Browsing data set	
	Accuracy	Macro- F_1	Accuracy ⁺	ma F_1
ULMFIT	0.922	0.892	0.564	0.431
ULMFIT mix-domain	–	–	0.508 (-9.9%)	0.433 (+0.4%)
ULMFIT cross-domain	–	–	0.665 (+17.9%)	0.506 (+17.4%)
STM	0.544	0.527	0.794	0.625
STM cross-domain	–	–	0.814 (+2.5%)	0.647 (+3.5%)

the classifier with a different data set. We also observed a higher perplexity when fine-tuning the language model using the browsing-unsup data set compared to using the news-crawl data set, indicating that the browsing data set might be more different from the WikiText-103 data set (Merity *et al.* 2016), where the language model was pre-trained.

On the other hand, the cross-domain method improved the performance of ULMFiT drastically with a more than 17% improvement in both accuracy and macro F_1 . This demonstrated the importance of in-domain labelled documents for supervised learning methods and that self-training can effectively bootstrap an in-domain data set from an out-of-domain classifier without any manual labelling. However, ULMFiT model still lagged behind the original STM model without domain adaptation.

6. Conclusions and future work

In this work, we mitigated the lack of accurate in-domain documents for text classification by proposing a novel two-stage approach. We first mine keywords from a noisy out-of-domain corpus and then use the keywords and unlabelled documents to train a dataless classifier. The proposed approach outperformed various supervised learning and dataless classification baselines by a large margin on a corpus of user browsing data set. By tapping on unlabelled in-domain documents, the model yields another 3% performance gain.

During the experiments, we identified that the multi-label problem is one of the main reasons why supervised learning methods failed. The proposed dataless method does not exploit the document labels and is thus more robust. In future work, we plan to explicitly model the multi-label classification problem and identify the segments of the document which represent different topics.

Acknowledgement. The first author was supported by the scholarship from ‘The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship’ and also ‘The 90th Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund)’. We would like to thank the anonymous reviewers for their careful reading of the manuscript and constructive criticism. We would like to thank our colleagues Bao-Dai Nguyen-Hoang and Khanh Huynh for helping to prepare the data sets and Akshay Bhatia for contributing one baseline for this work.

References

- Breve F.A., Zhao L. and Quiles M.G. (2010). Semi-supervised learning from imperfect data through particle cooperation and competition. In *Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN)*, Barcelona, Spain. Institute of Electrical and Electronics Engineers, pp. 1–8.
- Brodley C.E., Friedl M.A. et al. (1996). Identifying and eliminating mislabeled training instances. In *Proceedings of the National Conference on Artificial Intelligence*, Portland, Oregon. Association for the Advancement of Artificial Intelligence, pp. 799–805.

- Carbonell J.G. and Goldstein J.** (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, vol. 98. Association for Computing Machinery, pp. 335–336.
- Chang M.W., Ratinov L.A., Roth D. and Srikumar V.** (2008). Importance of semantic representation: Dataless classification. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, Chicago, Illinois, vol. 2. Association for the Advancement of Artificial Intelligence, pp. 830–835.
- Charoenphakdee N., Lee J., Jin Y., Wanvarie D. and Sugiyama M.** (2019). Learning only from relevant keywords and unlabeled documents. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pp. 3984–3993.
- Dahlmeier D.** (2017). On the challenges of translating NLP research into commercial products. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver, Canada, vol. 2. Association for Computational Linguistics, pp. 92–96.
- Dieterich T.G.** (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning* 40(2), 139–157.
- Druck G., Mann G. and McCallum A.** (2008). Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, Singapore. Association for Computing Machinery, pp. 595–602.
- Frénay B. and Verleysen M.** (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems* 25(5), 845–869.
- Gabrilovich E., Markovitch S. et al.** (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, Hyderabad, vol. 7. International Joint Conferences on Artificial Intelligence, pp. 1606–1611.
- Gamberger D., Lavrac N. and Groselj C.** (1999). Experiments with noise filtering in a medical domain. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999)*, Bled, Slovenia. International Machine Learning Society, pp. 143–151.
- Ganin Y., Ustinova E., Ajakan H., Germain P., Larochelle H., Laviolette F., Marchand M. and Lempitsky V.** (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17(1), 2096–2030. ISSN 1532-4435.
- Gerlach R. and Stamey J.** (2007). Bayesian model selection for logistic regression with misclassified outcomes. *Statistical Modelling* 7(3), 255–273.
- Howard J. and Ruder S.** (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 328–339.
- Hsu P.L. and Robbins H.** (1947). Complete convergence and the law of large numbers. *Proceedings of the National Academy of Sciences of the United States of America* 33(2), 25.
- Jin Y., Wanvarie D. and Le P.** (2017). Combining lightly-supervised text classification models for accurate contextual advertising. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Taipei, Taiwan, vol. 1. Asian Federation of Natural Language Processing, pp. 545–554.
- King B. and Abney S.P.** (2013). Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, USA. Association for Computational Linguistics, pp. 1110–1119.
- Krizhevsky A., Sutskever I. and Hinton G.E.** (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, Lake Tahoe, Nevada, USA, pp. 1097–1105.
- Lang K.** (1995). Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, Tahoe City, California, USA. International Machine Learning Society, pp. 331–339.
- Le Q. and Mikolov T.** (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning - Volume 32*, Beijing, China. International Machine Learning Society, pp. 1188–1196.
- Li C., Chen S., Xing J., Sun A. and Ma Z.** (2018a). Seed-guided topic model for document filtering and classification. *ACM Transactions on Information Systems (TOIS)* 37(1), 9.
- Li C., Xing J., Sun A. and Ma Z.** (2016). Effective document labeling with very few seed words: A topic model approach. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, Indianapolis, USA. Association for Computing Machinery, pp. 85–94.
- Li C., Zhou W., Ji F., Duan Y. and Chen H.** (2018b). A deep relevance model for zero-shot document filtering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 2300–2310.
- Li X. and Yang B.** (2018). A pseudo label based dataless naive bayes algorithm for text classification with seed words. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA. Association for Computational Linguistics, pp. 1908–1917.

- Maas A.L., Daly R.E., Pham P.T., Huang D., Ng A.Y. and Potts C.** (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Portland, Oregon, USA. Association for Computational Linguistics, pp. 142–150.
- Meng Y., Shen J., Zhang C. and Han J.** (2018). Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, Turin, Italy. Association for Computing Machinery, pp. 983–992.
- Merity S., Xiong C., Bradbury J. and Socher R.** (2016). Pointer sentinel mixture models. arXiv preprint [arXiv:1609.07843](https://arxiv.org/abs/1609.07843).
- Mudinas A., Zhang D. and Levene M.** (2018). Bootstrap domain-specific sentiment classifiers from unlabeled corpora. *Transactions of the Association of Computational Linguistics* **6**, 269–285.
- Nam J., Menca E.L. and Fürnkranz J.** (2016). All-in text: Learning document, label, and word representations jointly. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA. Association for the Advancement of Artificial Intelligence.
- Nam J., Menca E.L., Kim H.J. and Fürnkranz J.** (2017). Maximizing subset accuracy with recurrent neural networks in multi-label classification. In *Advances in Neural Information Processing Systems*, Long Beach, CA, USA. Curran Associates, pp. 5413–5423.
- Nettleton D.F., Orriols-Puig A. and Fornells A.** (2010). A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review* **33**(4), 275–306.
- Nguyen-Hoang B.D., Pham-Hong B.T., Jin Y. and Le P.** (2018). Genre-oriented web content extraction with deep convolutional neural networks and statistical methods. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation (PACLIC 32)*, Hong Kong, China. Association for Computational Linguistics, pp. 452–459.
- Pan S.J. and Yang Q.** (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* **22**(10), 1345–1359.
- Pappas N. and Henderson J.** (2019). Gile: A generalized input-label embedding for text classification. *Transactions of the Association for Computational Linguistics* **7**, 139–155.
- Peters M., Neumann M., Iyyer M., Gardner M., Clark C., Lee K. and Zettlemoyer L.** (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana. Association for Computational Linguistics, pp. 2227–2237. doi: [10.18653/v1/N18-1202](https://doi.org/10.18653/v1/N18-1202).
- Ribeiro M.T., Singh S. and Guestrin C.** (2016). “why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA. Association for Computing Machinery, pp. 1135–1144.
- Sachan D., Zaheer M. and Salakhutdinov R.** (2018). Investigating the working of text classifiers. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA. Association for Computational Linguistics, pp. 2120–2131.
- Settles B.** (2011). Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland. Association for Computational Linguistics, pp. 1467–1478.
- Song Y. and Roth D.** (2014). On dataless hierarchical text classification. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Quebec, Canada. Association for the Advancement of Artificial Intelligence Press.
- Song Y., Upadhyay S., Peng H., Mayhew S. and Roth D.** (2019). Toward any-language zero-shot topic classification of textual documents. *Artificial Intelligence* **274**, 133–150.
- Song Y., Upadhyay S., Peng H. and Roth D.** (2016). Cross-lingual dataless classification for many languages. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, New York, USA. International Joint Conferences on Artificial Intelligence, pp. 2901–2907.
- Sun J.W., Zhao F.Y., Wang C.J. and Chen S.F.** (2007). Identifying and correcting mislabeled training instances. In *Future Generation Communication and Networking (FGCN 2007)*, Jeju-Island, Korea, vol. 1. Institute of Electrical and Electronics Engineers, pp. 244–250.
- Swartz T.B., Haitovsky Y., Vexler A. and Yang T.Y.** (2004). Bayesian identifiability and misclassification in multinomial data. *Canadian Journal of Statistics* **32**(3), 285–302.
- Wang S. and Manning C.D.** (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, Jeju Island, Korea. Association for Computational Linguistics, pp. 90–94.
- Wang X., Wei F., Liu X., Zhou M. and Zhang M.** (2011). Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, Glasgow, Scotland. Association for Computing Machinery, pp. 1031–1040.
- Yin W., Hay J. and Roth D.** (2019). Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pp. 3905–3914.

Yogatama D., Dyer C., Ling W. and Blunsom P. (2017). Generative and discriminative text classification with recurrent neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, Sydney, Australia. International Machine Learning Society.

Zha D. and Li C. (2019). Multi-label dataless text classification with topic modeling. *Knowledge and Information Systems* 61(1), 137–160.

Zhang X., Zhao J. and LeCun Y. (2015). Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, Montreal, Canada. Curran Associates, pp. 649–657.

Zheng R., Tian T., Hu Z., Iyer R., Sycara K. et al. (2016). Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japanpp. The COLING 2016 Organising Committee, pp. 2678–2688.

Zipf G.K. (1949). *The Principle of Least Effort: An Introduction to Human Ecology*. Boston, USA: Addison Wesley.

A. Appendix

A.1 Automatically mined keywords at different label noise rates

Table A1. Keywords for the category ‘talk.politics.mideast’ with 0.1/0.4/0.7 label noise

ϵ	PMI	PMI-FREQ	MMR	FREQ	MI
0.1	diaspora	arab	arab	israel	govt
	sarajevo	israel	israeli	jews	agnostic
	karabag	israeli	arabs	war	suspects
	millet	muslims	serdar	muslims	infer
	rabin	arabs	argic	jewish	testify
	gaza	jews	israel	arab	excuses
	shostack	serdar	armenian	muslim	reminded
	agdam	argic	turkish	during	arrogance
	davidsson	muslim	turks	state	salem
	barbarism	jewish	armenians	history	examine
0.4	yitzhak	muslims	armenian	israel	inquisition
	hamas	israel	arab	war	paradox
	azerbaijanis	armenian	argic	jews	agnostic
	settlements	muslim	genocide	muslims	ponder
	balkan	jewish	serdar	jewish	reward
	lehi	jews	arabs	during	dragging
	erzurum	arab	armenians	muslim	affair
	gaza	argic	serbs	state	thy
	plo	genocide	israel	arab	overlooked
	azerbaijan	serdar	armenia	actually	prophecies

Table A1. Continued.

€	PMI	PMI-FREQ	MMR	FREQ	MI
0.7	iraqis	muslim	armenians	jews	<u>fundamentally</u>
	<u>bayonets</u>	armenians	<u>serdar</u>	israel	<u>mittchell</u>
	plo	<u>serdar</u>	<u>argic</u>	<u>ever</u>	<u>elvis</u>
	<u>memoirs</u>	<u>argic</u>	muslim	<u>state</u>	<u>meters</u>
	<u>arf</u>	arab	moslem	<u>others</u>	<u>tricky</u>
	azeris	moslem	armenian	<u>during</u>	<u>cent</u>
	moslem	armenian	turkish	<u>actually</u>	<u>holland</u>
	<u>asala</u>	turkish	jew	<u>put</u>	<u>affects</u>
	<u>sdpa</u>	jew	arabs	<u>group</u>	<u>mysteries</u>
	<u>exterminated</u>	jews	genocide	<u>give</u>	<u>pan</u>

Table A2. Keywords for the category 'rec.autos' with 0.1/0.4/0.7 label noise. The ambiguous keywords are underlined

€	PMI	PMI-FREQ	MMR	FREQ	MI
0.1	geico	car	cars	car	car
	camaro	cars	car	cars	engine
	corvette	ford	ford	engine	cars
	rotors	engine	mustang	ford	<u>insurance</u>
	diesels	mustang	camaro	<u>buy</u>	<u>clutch</u>
	mustang	nissan	geico	<u>price</u>	<u>manual</u>
	sunroof	engines	nissan	<u>miles</u>	wheel
	tach	camaro	diesels	<u>big</u>	<u>design</u>
	nissan	geico	corvette	<u>speed</u>	<u>sports</u>
	shifter	suspension	suspension	<u>put</u>	<u>bought</u>
0.4	diesels	car	car	car	car
	corvette	cars	cars	cars	cars
	odometer	ford	ford	engine	ford
	lexus	engine	mustang	<u>price</u>	engines
	camaro	engines	camaro	<u>actually</u>	<u>brake</u>
	audi	mustang	<u>cylinder</u>	<u>buy</u>	wheels
	convertible	<u>exhaust</u>	<u>tranny</u>	<u>every</u>	<u>clutch</u>
	<u>liter</u>	camaro	coupe	<u>big</u>	wheel

Table A2. Continued.

€	PMI	PMI-FREQ	MMR	FREQ	MI
	geico	<u>cylinder</u>	diesels	<u>put</u>	mph
	<u>tranny</u>	<u>tranny</u>	engine	<u>miles</u>	<u>design</u>
0.7	traction	engine	<u>gt</u>	car	engine
	convertible	cars	mustang	<u>little</u>	ford
	wagon	<u>gt</u>	engine	<u>best</u>	<u>cult</u>
	<u>gt</u>	ford	wagon	<u>called</u>	<u>alot</u>
	mustang	car	traction	<u>every</u>	camaro
	mazda	mustang	cars	<u>price</u>	mustang
	<u>seats</u>	wagon	<u>seats</u>	engine	<u>raised</u>
	ford	traction	convertible	cars	<u>represent</u>
	<u>exhaust</u>	<u>seats</u>	car	<u>big</u>	<u>conversion</u>
	<u>pulse</u>	convertible	ford	<u>probably</u>	<u>taxes</u>

Table A3. Keywords for the category 'rec.sport.baseball' with 0.1/0.4/0.7 label noise. The ambiguous keywords are underlined

€	PMI	PMI-FREQ	MMR	FREQ	MI
0.1	<u>royals</u>	pitcher	pitcher	baseball	<u>team</u>
	batters	baseball	batting	<u>games</u>	<u>games</u>
	<u>alomar</u>	batting	pitching	<u>team</u>	baseball
	<u>larkin</u>	pitching	pitches	<u>hit</u>	<u>season</u>
	<u>sandberg</u>	pitches	batters	<u>players</u>	<u>players</u>
	<u>platoon</u>	hitter	hitter	pitcher	<u>player</u>
	<u>ws</u>	batter	batter	<u>league</u>	pitcher
	<u>mattingly</u>	braves	braves	<u>season</u>	<u>win</u>
	<u>boggs</u>	jays	jays	<u>lot</u>	<u>league</u>
	<u>sabo</u>	<u>sox</u>	<u>sox</u>	<u>play</u>	hitting
0.4	yankees	pitcher	pitcher	baseball	<u>team</u>
	<u>hirschbeck</u>	hitter	hitter	<u>team</u>	pitcher
	<u>phillies</u>	batter	batter	<u>hit</u>	<u>player</u>
	<u>padres</u>	<u>sox</u>	<u>sox</u>	<u>games</u>	<u>win</u>
	orioles	jays	jays	pitcher	<u>season</u>

Table A3. Continued.

€	PMI	PMI-FREQ	MMR	FREQ	MI
	mets	batting	batting	player	hitter
	bosio	baseball	pitchers	best	games
	sabo	pitchers	yankees	probably	batter
	pitched	yankees	hirschbeck	players	teams
	rbi	hirschbeck	ball	ball	batting
0.7	dl	batting	batting	probably	inning
	winfield	hitting	jays	little	cult
	martinez	jays	batter	day	stats
	umpires	batter	bat	lot	innings
	sabo	bat	pitcher	try	pitches
	batting	baseball	morris	kind	player
	hirschbeck	pitcher	inning	enough	season
	jays	morris	umpires	actually	pitch
	batter	inning	hitter	post	symptoms
	inning	umpires	stats	give	homosexuals

B. Appendix

B.1 Construction of news-crawl data sets

In this section, we describe the method we used to crawl the labelled data sets from newswire sites without manually labelling the articles so that researchers can reproduce our results or create data sets for other categories.

Many websites, especially newswire sites, categorise their content into a list of predefined categories. An example is shown in Figure B1.

Some of these sites encode the category name in the URL like the case in Figure B2. We can apply regular expressions on the URLs to extract the category of the article. An example is shown in Table B1.



Figure B1. Screenshot from The New York Times homepage.

Table B1. Sample regular expression to extract the category of news articles

Regular expression for URL	Category
<code>nytimes.com/([0-9]*)/([0-9]*)/([0-9]*)/(arts books movies theater)</code>	Arts and entertainment
<code>pethelpful.com/(rabbits dogs birds cats misc)</code>	Pets

Table B2. Domains from where the news-crawl data sets were crawled

News-crawl data set	News-crawl-v2 data set
reuters.com	cooking.nytimes.com
nytimes.com	nytimes.com
theguardian.com	instructables.com
independent.co.uk	community.babycenter.com
thestar.com.my	pethelpful.com
alphamom.com	olx.ph
ultimate-guitar.com	biblegateway.com
dailynews.com	coupons.com
dallasnews.com	groupon.com
cheatsheet.com	shopback.sg
dailyfinance.com	psychologytoday.com
pawnation.com	independent.co.uk
religionnews.com	marriage.com
	helpguide.org



Figure B2. Sample URL and its different components.

The other websites which do not include the category in the URL usually have a list page where we can crawl the list of URLs related to a category, such as www.reuters.com/news/archive/entertainmentNews and <https://www.dw.com/en/auto-industry/t-17282970>. Tabel B2 shows the full list of domains where *news-crawl data set* and *news-crawl-v2 data set* are crawled.