Iterative Seed Word Selection for Weakly-Supervised Text Classification with Bayesian Error Estimation

Jin Yiping
Senior Research Scientist - NLP
Knorex Pte. Ltd., Singapore
jinyiping@knorex.com

Akshay Bhatia
Research Scientist - NLP
Knorex Pte. Ltd., Singapore
akshay.bhatia@knorex.com

Dittaya Wanvarie
Assistant Professor
Faculty of Science, Chulalongkorn University
Dittaya.W@chula.ac.th

Abstract

Weakly-supervised text classification aims to induce text classifiers from only a few user-provided seed words. Regardless of the underlying model, the quality of the seed words has a significant impact on the classification accuracy. The vast majority of previous work assumes high-quality seed words are given. However, the expert-annotated seed words are non-trivial to come up with. Furthermore, in the weakly-supervised learning setting, we do not have access to any labeled document, neither for training nor for validation. Therefore, there is no way to detect and eliminate bad seed words to improve classification accuracy. In this work, we remove the need for expert-annotated seed words by firstly mining (noisy) candidate seed words associated with the category names. We then train interim models with individual candidate seed words in an iterative manner. Lastly, we use the Bayesian error estimation method to estimate the interim models' error rate in an unsupervised manner. The keywords that yield the lowest estimated error rates are added to the final seed word set. A comprehensive evaluation of six binary classification tasks on four popular datasets demonstrates the effectiveness of the proposed method. It outperforms a baseline using only the category name as seed word and obtained comparable performance as a counterpart using expert-annotated seed words.

1 Introduction

Weakly-supervised text classification eliminates the need for any labeled document and induces classifiers with only a handful of carefully chosen seed words (Meng et al., 2018). Recent works demonstrated that weakly-supervised models could sometimes achieve comparable if not better performance than fully-supervised baselines (Mekala and Shang, 2020; Jin et al., 2020). However, some researchers pointed out that the choice of initial seed words has a significant impact on the performance of weakly-supervised models (Li et al., 2018; Jin et al., 2020). The vast majority of previous work assumed high-quality seed words are given. However, many seed words reported in previous work are not intuitive to come up with. For example, in Meng et al. (2019), the seed words used for the category "Soccer" are {cup, champions, united} instead of more intuitive keywords like "soccer" or "football". We conjecture the authors might have tried these more general keywords but avoided them because they do not perform as well on the test set.

While it is common to use labeled corpora to evaluate weakly-supervised text classifiers in the literature, we do not have access to any labeled document for new categories in the real-world setting. Therefore, there is no way to measure the model's performance and to detect bad seed words to improve classification accuracy. A similar concern on assessing active learning performance at runtime has been raised by Kottke et al (2019).

In this work, we device $OptimSeed$, a novel framework to automatically compose and select seed words for weakly-supervised text classification. We firstly mine (noisy) candidate seed words associated with the category names. We then train interim models with individual candidate seed words in an iterative manner. Lastly, we use an unsupervised error estimation method to estimate the interim models' error rates. The keywords that yield the lowest estimated error rates are selected as the final seed word set. A comprehensive evaluation of six classification tasks on four popular datasets demonstrates the effectiveness of the proposed method. The proposed method outperforms a baseline using only the category name as seed word and obtained comparable performance as a counterpart using expert-annotated seed words. We use binary classification as a case study in this work while the idea can be generalized to multi-class classification using one-vs.-rest strategy.

The contributions of this work are three-fold:

1. To our best knowledge, this is the first work using unsupervised error estimation to improve weakly-supervised text classification's performance. The two fit perfectly because neither requires any labeled document.
2. We conduct an in-depth study on the impact of different seed words on weakly-supervised text classification, supported by experiments with various models and classification tasks.
3. The proposed method generates keyword sets that yield consistent and competitive performance against various baselines and expert-annotated seed words.

2 Related Work

We review the literature in three related fields: (1) weakly-supervised text classification, (2) unsupervised error estimation, and (3) keyword mining.

2.1 Weakly-Supervised Text Classification

Weakly-supervised text classification (Druck et al., 2008; Jin et al., 2017; Meng et al., 2018, 2019) aims to use a handful of labeled seed words to induce text classifiers instead of relying on labeled documents.

Traditionally, the labeled seed words are used to provide pseudo-labels to the unlabeled documents, either by counting the occurrences of seed words or calculating the cosine similarity between the documents and the seed words. The pseudo-labeled documents are then used to perform either supervised or semi-supervised learning (Liu et al., 2004; Charoenphakdee et al., 2019). An obvious drawback of this approach is the noise introduced by pseudo labeling. Druck et al. (2008) addressed this problem by proposing generalized expectation (GE), which specifies the expected posterior probability of labeled seed words appearing in each category. For example, if the word "puck" is a labeled seed word for the category "hockey", the word is expected to occur 90% in documents belonging to the category "hockey" and 10% in a document belonging to another category. GE is trained by optimizing towards satisfying the posterior constraints without making use of pseudo-labeled documents.

Chang et al. (2008) proposed the first embedding based weakly-supervised text classification method. They mapped category names and documents into the same semantic space whose dimensions correspond to Wikipedia concepts. Document classification is then performed by searching for the nearest category embedding given an input document.

Meng et al. (2018) proposed weakly-supervised neural text classification. Their method composes of a pseudo-document generation process that generates unambiguous pseudo-documents instead of assigning pseudo-labels to real documents. The pseudo-documents are used to induce neural text classifiers with different architectures such as convolutional neural networks (Kim, 2014) or Hierarchical Attention Network (Yang et al., 2016).

Most recently, Mekala and Shang (Mekala and Shang, 2020) addressed the ambiguity of seed words by explicitly learning different senses of each word with contextualized word embeddings. They first performed k-means clustering for each word in the vocabulary to identify potentially different senses, then eliminated the ambiguous keyword senses with the assumption that most labeled keywords occur in documents belonging to the target category. They demonstrated that the method outperformed counterparts without disambiguation or using a pre-trained word sense disambiguation model. They also expanded the seed keywords automatically by adding new words whose embeddings are near the seed words. Mekala and Shang (Mekala and Shang, 2020) is closest to our work. However, they select new seed words based on similarity while we select new seed words based on the estimated performance.

2.2 Unsupervised Error Estimation

Unsupervised error estimation is a critical yet understudied problem. It aims to estimate the error rate/accuracy of a single classifier or a list of classifiers *without a labeled evaluation dataset*. It is widely relevant to machine learning models in production, such as when a pre-trained model is applied to a new domain or when labeled dataset is costly to obtain. Likewise, unsupervised error estimation is crucial to weakly-supervised classification because we do not have access to any labeled document and cannot calculate traditional evaluation metrics like accuracy or F_1 score. Unfortunately, no previous work in weakly-supervised classification considered this aspect. They all trained classifiers without labeled *training* datasets but evaluated used labeled *evaluation* datasets.

Most work in unsupervised error estimation aims to derive the error rate analytically by making certain simplifying assumptions. Donmez et al. (2010) and Jaffe et al. (2015) assumed the marginal probability of the category $p(y)$ is known and derived the risks $R(f_1), ..., R(f_k)$ of each of the k predictors. Platanios et al. (2014) assumed classifiers make conditionally independent errors and derived the relationship between the error rate and the pairwise classifier agreement. While these approaches laid an important theoretical foundation, most assumptions cannot be met for real-world datasets and classifiers.

Platanios et al. (2016) proposed a Bayesian approach for error estimation. The true category is modeled as a latent variable, and the method used a generative process to generate each classifier's predictions based on the true latent category and the latent error rate. The approach was benchmarked with various baselines such as majority vote and Platanios et al. (2014) and achieved superior performance. The estimated accuracy is usually within a few percents from the true accuracy.

2.3 Keyword Mining

Keyword mining aims to bootstrap high-quality keyword lexicons from a small set of seed words. It has been widely used in mining opinion lexicons (Hu and Liu, 2004; Hai et al., 2012) and technical glossaries (Elhadad and Sutaria, 2007). Hu and Liu (2004) and Riloff and Wiebe (2003) learned part-of-speech or dependency parsing patterns from seed words and unlabeled corpora, then used these patterns to bootstrap words similar to the seed words. Jin et al. (2020) used $pmi\text{-}freq$, a modified point-wise mutual information (PMI) to mine keywords from noisily labeled corpora.

We want to draw the association between keyword mining and weakly-supervised text classification. Both tasks take a small list of seed words and unlabeled corpus as input, aiming to "expand" the knowledge about the target semantic category. Having more high-quality keywords will improve classification accuracy while an accurate classifier will make the keyword mining task much easier by eliminating irrelevant/noisy documents.

3 Method

Figure 1 overviews OptimSeed, a framework to select seed words for weakly-supervised text classification involving the following steps: (1) expanding candidate keywords from a single seed word, (2) training interim classifiers with individual candidate seed keywords using weakly supervision, (3) select the final seed words with the feedback from unsupervised error estimation. We discuss the proposed framework in detail in the following sections. To make our paper self-contained, we will also brief the weakly-supervised classification model and the unsupervised error estimation model used in this work.

3.1 Expanding Candidate Keywords from a Single Seed

We purposely avoid expert-annotated seed words and use either the category name or trivial keywords (e.g., "good" and "bad" for sentiment classification tasks) as the only input seed word. We use a keyword expansion algorithm to mine more candidate keywords. Since the input seed word is not perfectly discriminative (e.g., "world" can occur in the context of politics or "world cup"), any keyword expansion algorithm will unavoidably introduce some noise. However, it is still better than relying on the single seed word alone (as demonstrated in experiment section).

We use $pmi\text{-}freq$ (Equation 1) to mine associated keywords following Jin et al. (2020). It is a product of the logarithm of the candidate keyword w's document frequency and the point-wise mutual information between the candidate keyword and the seed word s.

$$pmi\text{-}freq(w; s) \equiv log\, df(w) log \frac{p(w, s)}{p(w)p(s)} \quad (1)$$

Additionally, we filter the mined keywords based on its part-of-speech tag depending on the classification task. We keep only noun candidates for topic classification and adjective candidates for sentiment classification.

Overlapping keywords might occur in semantically similar categories. For example, when classifying between "baseball" and "hockey", keywords like "season" or "player" might be mined for both categories. However, we prefer more discriminative keywords, which can help the classifier better distinguish the categories. We apply maximal marginal relevance (MMR) (Carbonell and Goldstein, 1998) to re-rank the candidate keywords. MMR was originally used in information retrieval to ensure the retrieved set of documents are both relevant and diverse.

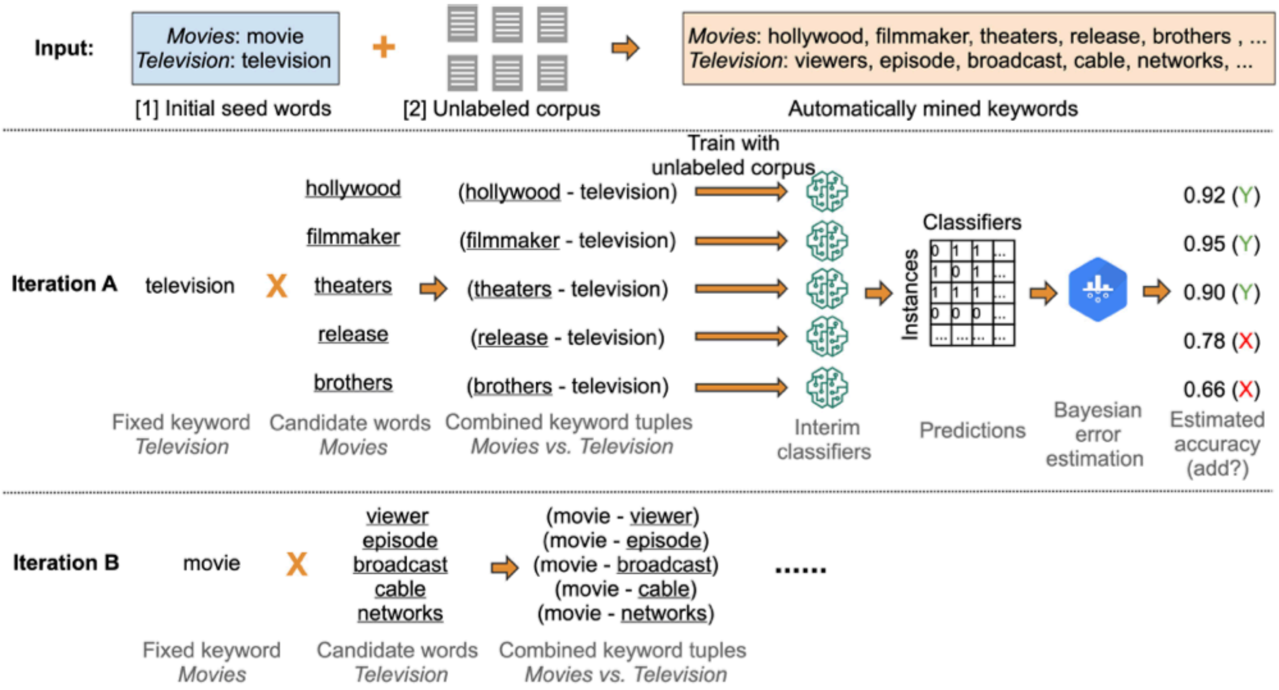

Figure 1: OptimSeed, a method to select seed words for weakly-supervised text classification. We fix the seed word for one category in each iteration and combine it with each candidate word in the other category. Each keyword tuple is used to train a separate interim classifier, whose predictions are used in Bayesian error estimation.

$$mmr(w, s_m) \equiv \lambda \overbrace{pmi\text{-}freq(w; s_m)}^{\text{T1}} \\ \underbrace{- (1-\lambda) \max_{m \neq n} pmi\text{-}freq(w; s_n)}_{\text{T2}} \quad (2)$$

In our context, MMR is defined in Equation 2. While the first term (T1) measures the association between the candidate word w and the category seed word s_m, the second term (T2) measures the association between w and the seed word from another category. λ controls the contribution of the two terms, and we use a default $\lambda = 0.5$ to give them an equal contribution. Intuitively, MMR score will be high when T1 is high, and T2 is low.

3.2 Training interim classifiers

The candidate keywords and unlabeled dataset are used to induce interim classifiers, as demonstrated in Figure 1. Specifically, we keep the seed word for one category fixed at each iteration and use it in conjunction with each candidate seed word from the other category. We train one interim classifier for each such seed word combination. We then obtain each interim classifier's predictions on the *unlabeled* test set [1], which provides the input for unsupervised error estimation described in Section 3.3.

We use Generalized Expectation (GE) (Druck et al., 2008) to train both interim classifiers and the final classifier (using all selected keywords) because of its competitive performance and fast training speed [2].

GE provides training signals in terms of labeled keywords instead of labeled documents. The keywords are translated into constraint functions such as $puck \rightarrow baseball : 0.1, hockey : 0.9$, which means the keyword "puck" is expected to occur 90% of the times in a document belonging to the category "hockey" while 10% in a document belonging to the category "baseball".

Each constraint k is translated into a term in the objective function in Equation 3 and the underlying logistic regression model is trained by minimizing the distance between the reference distribution $\hat{p}(y|x_k > 0)$ (specified by the constraint function) and the empirical distribution $\tilde{p}(y|x_k > 0)$ (predicted by the model).

$$\mathcal{O} = -\sum_{k \in K} dist(\hat{p}(y|x_k > 0) || \tilde{p}(y|x_k > 0)) \quad (3)$$

[1] The use of unlabeled test set is similar to a majority vote ensemble. It does not cause "training on test data".

[2] All GE models in this work can be trained within a few seconds using a single CPU core.

3.3 Keyword Evaluation with Bayesian Error Estimation

We apply unsupervised error estimation based on the interim classifiers' predictions to estimate their accuracy and use it as a guidance to select the best seed words for the final classifier. We use the Bayesian error estimation (BEE) model (Platanios et al., 2016) to perform this step because it achieves state-of-the-art performance with the estimated accuracy within a few percents of the true accuracy. In BEE, each instance's true label is latent, while each model's predictions are observed. The accuracy/error rate can be easily derived from the predictions and the latent true labels.

BEE defines the following generative process:

1. Draw p ~Beta(α_p,β_p) representing the prior probability for the true label to be 1.
2. For each data example x_i, draw a true label $l_i \sim$ Bernoulli(p).
3. For each classifier, draw an error rate e_j ~Beta(α_e,β_e).
4. Draw an output label, $\hat{f}_{ij}$ for the i^{th} example from the j^{th} classifier following the distribution:
$$\hat{f}_{ij} = \begin{cases} l_i, & \text{with probability} 1 - e_j \\ 1 - l_i, & \text{otherwise} \end{cases}$$

The inference of BEE is performed using Gibbs sampling. We refer readers to Platanios et al. (2016) for further details.

In our experiment, performing BEE alone often leads to an unstable result; sometimes, the estimated accuracy is 30-40% away from the true accuracy. A more thorough error analysis reveals that the model often fails when multiple classifiers (almost) always predict either negative or positive. Since all unsupervised error estimation methods rely on agreements between classifiers to some extent, if two or more classifiers always predict negative, the model will observe an artificially high agreement, causing it to believe the true labels for all examples are negative.

We propose a simple method to address this problem: we calculate the empirical marginal distribution of the category $\hat{p}(y)$ of each interim classifier and calculate their mean and variance. We eliminate outlier classifiers whose $\hat{p}(y)$ lie outside one standard deviation from the mean before performing BEE. While some previous work assumed the true marginal probability of the category $p(y)$ is available (which is not the case for most real-world problems), we base solely on the empirical distribution, which can be easily observed.

4 Experiment

In this section, we conduct a comprehensive evaluation of our framework on various datasets and classification tasks.

4.1 Datasets

We use six binary classification tasks from four datasets to evaluate our framework. We choose the evaluation tasks so that they cover different granularities and domains. Table 1 provides the dataset statistics, and the details are as follows:

- **AG's News Dataset:** [3] contains 120,000 documents evenly distributed into 4 coarse categories. We use two binary classification tasks: "Politics" vs. "Technology" and "Business" vs. "Sports".
- **The New York Times (NYT) Dataset:** [4] contains 13,081 news articles covering 5 coarse and 25 fine-grained categories. We use two fine-grained binary classification tasks that we think are the most difficult: "International Business" (InterBiz) vs. "Economy" and "Movies" vs. "Television".
- **Yelp Restaurant Review Dataset:** [5] contains 38,000 reviews evenly distributed into 2 categories: "Positive" vs. "Negative".
- **IMDB Movie Review Dataset:** [6] contains 50,000 reviews evenly distributed into 2 categories: "Positive" vs. "Negative".

4.2 Baselines

We report the performance of our framework and the following weakly-supervised baselines using different sets of seed words:

- **Dataless (Chang et al., 2008):** [7] maps both input documents and category seed words

[3] https://github.com/mhjabreel/CharCnn_Keras/tree/master/data/ag_news_csv

[4] https://github.com/yumeng5/WeSHClass/tree/master/nyt

[5] https://github.com/yumeng5/WeSTClass/tree/master/yelp

[6] https://ai.stanford.edu/~amaas/data/sentiment

[7] https://github.com/yqsong/Dataless Classification

Categories	**# Train**	**# Test**	**Avg Words**
Politics-Tech	60,000	3,800	45
Business-Sports	60,000	3,800	46
InterBiz-Econ	569	164	745
Movie-TV	432	90	1,033
Yelp +/-	30,400	7,600	155
IMDB +/-	25,000	25,000	273

Table 1: Dataset statistics.

into a semantic space using Explicit Semantic Analysis (ESA) (Gabrilovich et al., 2007) over Wikipedia concepts and assigns the category nearest to the input document's embedding.

- **MNB/Priors (Settles, 2011):** [8] increases priors for labeled keywords in a Naïve Bayes model and learns from an unlabeled corpus using EM algorithm.
- **WeSTClass (Meng et al., 2018):** [9] weakly-supervised neural text classifier trained using pseudo documents. We use the CNN architecture as recommended in the original paper.
- **ConWea (Mekala and Shang, 2020):** [10] leverages contextualized word representations to differentiate multiple senses. It also trains classifiers and expands seed words in an iterative manner.

For each baseline, we use the implementation provided by the original authors and follow the parameters in the paper except (1) for Dataless model, we use a more recent Wikipedia dump downloaded on 10 December 2019, and (2) for ConWea, we use a batch size of 128 instead of 256 because of the memory limit of our GPU (8GB).

We also report the performance of **LR**, a supervised logistic regression model trained using all the documents in the training set for comparison [11].

4.3 Experiment Settings

In all experiments, we mine 16 candidate seed words for each category [12]. We perform Bayesian error estimation once for each category and add the top candidate words in a batched manner. We select a candidate keyword for the final classifier if its estimated accuracy is among the top three or is higher than 0.9 [13]. For GE, we use a reference distribution of 0.9 (meaning a labeled keyword is expected to appear in its specified categories 90% of the times) following Druck et al. (2008).

Table 2 shows the seed words used in our work as well as in previous work [14]. It is manifest that the seed words we use are much more trivial to come up. They may not be as discriminative as expert-picked keywords, making the task more challenging for the weakly-supervised models.

Class	**Ours**	**Previous Work**
Politics	political;	democracy religion liberal;
Tech	technology	scientists biological computing
Business	business;	economy industry investment;
Sports	sports	hockey tennis basketball
InterBiz	international;	china union euro;
Economy	economy	fed economists economist
Movies	movie;	hollywood directed oscar;
Television	television	episode viewers episodes
Yelp & IMDB	good;	terrific great awesome;
	bad	horrible subpar disappointing

Table 2: Seed words used in our work and in previous work.

4.4 Classification Performance

Table 3 shows each model's classification accuracy on the six classification tasks and Table 4 presents each model's average accuracy.

We can see that the Dataless model performed poorly compared to other models. The selection of keywords does not seem to have much impact on

[8] https://github.com/burrsettles/dualist

[9] https://github.com/yumeng5/WeSTClass

[10] https://github.com/dheeraj7596/ConWea

[11] We use the logistic regression implementation in scikit-learn with default parameters and tf-idf features.

[12] We will train an interim classifier for each seed word. We choose 16 seed words as a trade-off between the seed word quality and the computation.

[13] Mekala and Shang (2020) observed that three seed words per class are needed for reasonable performance while more high-quality keywords help. Therefore, we use the accuracy threshold of 0.9 to include additional keywords.

[14] The seed words for NYT corpus were reported in Meng et al. (2019) and the rest are from Meng et al. (2018). No previous work in weakly supervision used IMDB dataset, so we use the same manual seed words as Yelp dataset.

Method	**Politics-Tech**				**Business-Sports**				**InterBiz-Economy**			
	cate	**all**	**ours**	**gold**	**cate**	**all**	**ours**	**gold**	**cate**	**all**	**ours**	**gold**
Dataless	.501	**.536**	.514	.502	.500	**.513**	.502	.504	.591	**.762**	.750	.671
MNB/Priors	.873	**.894**	.889	.889	**.956**	.922	.939	.929	.585	.445	.543	**.939**
WeSTClass	.874	.885	**.895**	.888	.927	.896	**.948**	.943	.777	**.830**	**.830**	.751
ConWea	.715	.581	**.737**	.714	.391	.806	.670	**.820**	.751	.712	.712	**.843**
GE	.869	.876	.878	**.885**	**.930**	.723	**.930**	.794	.707	.793	.817	**.915**
lr	0.963				0.986				0.902			

Method	**Movies-Television**				**Yelp Review**				**IMDB Review**			
	cate	**all**	**ours**	**gold**	**cate**	**all**	**ours**	**gold**	**cate**	**all**	**ours**	**gold**
Dataless	.678	**.811**	.700	.678	.510	**.580**	.555	.522	.501	**.617**	.604	.522
MNB/Priors	.678	**.744**	.678	.689	.509	.609	**.715**	.517	.511	.534	**.540**	.503
WeSTClass	.504	**.844**	.766	.621	.783	.633	.588	**.815**	**.677**	.646	.606	.605
ConWea	.669	.727	**.770**	.764	.510	**.514**	.513	.507	.565	.582	.557	**.591**
GE	.944	.967	**.989**	.978	.680	.752	.752	**.793**	.696	**.741**	.722	.740
lr	0.855				0.922				0.883			

Table 3: Accuracy scores for all methods on six classification tasks. **cate**, **all**, **ours**, **gold** indicates the result using the category name, all seed words mined using keyword mining, keywords selected by OptimSeed and expert-composed keywords used in previous work. For each model-dataset combination, we highlight the best performance in bold.

Method	**cate**	**all**	**ours**	**gold**
Dataless	.547	**.637**	.604	.567
MNB/Priors	.685	.691	.717	**.744**
WeSTClass	.757	**.789**	.772	.770
ConWea	.600	.654	.660	**.707**
GE	.804	.809	.848	**.851**
lr	0.918			

Table 4: Average accuracy scores for all methods on all six classification tasks. The best-performing keyword set is highlighted in bold.

it. Using all automatically mined keywords always yielded the best performance, followed by using the keywords selected by OptimSeed.

GE obtained the best or close to the best performance for all seed word sets. The average accuracy of GE using OptimSeed seed words is only 0.3% lower than using expert-composed seed words, virtually eliminating human experts from the loop. It also performed 4.4% better than using the category name alone as seed words. We note that both the keyword expansion and Bayesian error estimation (BEE) steps are essential to performance improvement. While using all keywords may sometimes perform worse than using only the category names (e.g., Business-Sports), OptimSeed keywords almost always achieve better accuracy, demonstrating its ability to remove noisy keywords with the help of BEE. On average, GE+OptimSeed's accuracy is 7% below a fully-supervised logistic regression model trained on hundreds to tens of thousands of labeled documents.

The seed words selected by OptimSeed also improved the average accuracy of MNB/Priors, WeSTClass and ConWea by 3.2%, 1.5% and 6% separately compared to using only the category name as the seed word. It shows an opportunity to use our framework in conjunction with other weakly-supervised text classification methods [15].

OptimSeed seed words achieved the best performance for three out of six classification tasks for WeSTClass. However, its performance was very poor for the Yelp review dataset. We found out that WeSTClass wrongly associated many food names to the positive category because of the word "delicious" in our seed word set.

Surprisingly, ConWea performed lackluster compared with other baselines. While it claimed to resolve ambiguity through contextualized embeddings, we found the seed words ConWea automatically expanded to be much more ambiguous than OptimSeed. Table 5 shows the expanded keywords for the AGNews Business-Sports classification task using "cate" (acc: 0.39) and "gold" (acc: 0.82) key-

[15] We also tried using the same model for interim classifier & final classifier (e.g., if the final classifier is Dataless, we also train Dataless interim classifiers). However, the performance is no better than using GE interim classifiers.

words. Besides, ConWea requires huge disk space since it needs to store the contextualized word embeddings for each word occurrence in the corpus.

Cate (acc: 0.39)
new fullquote world stocks reuters target href HTTP ticket oil percent prices york wednesday
game season yesterday team Sunday million billion coach victory year company announced points win victory year
Gold (acc: 0.82)
billion ticker corp shares reuters company stocks percent york inc href oil HTTP said prices fullquote target
last first team second scored cup victory cup world win three game yesterday night season network sports Sunday series sox coach

Table 5: Keywords added by ConWea for "Business-Sports" with different initial seed words.

4.5 Case Study

To demonstrate the working of our proposed framework, we present a case study on the classification task "International Business" vs. "Economy" and show how the seed words for the category "economy" evolve. We can see that most keywords added by the keyword expansion algorithm are related to the category. However, some ambiguous keywords may occur in other contexts. The Bayesian error estimator (BEE) successfully identified top keywords such as "economist" and "economists" and eliminated poor-performing keywords like "unemployment" and "inflation". We note that BEE consistently over-estimated the accuracy. It is likely because some keywords may often occur together (e.g. "inflation" and "rate"), causing the induced interim classifiers to have a higher agreement level. Nevertheless, BEE allowed us to select a better seed word set, which improved the accuracy by 11% from the model using only the category name as seed word.

Stage:Acc	**Seed Words for "Economy"**
Init: 0.707	economy
Keyword Expansion: 0.793	*purchases* pace index *borrowing* unemployment economists *economy* stimulus rates recovery economist rate *fed* reserve inflation *growth*
Est. Acc (True Acc)	economist: 0.98 (0.81) economists: 0.96 (0.82) rate: 0.95 (0.78) recovery: 0.93 (0.76) index: 0.90 (0.79) pace: 0.89 (0.82) rates: 0.88 (0.75) reserve: 0.86 (0.81) inflation: 0.84 (0.70) stimulus: 0.82 (0.72) unemployment: 0.78 (0.64)
Final: 0.817	economist economists rate recovery index

Table 6: Seed words for "Economy" at different stages of the OptimSeed framework. The seed words in italics in the second row are removed as outliers.

5 Conclusion and Future Work

Weakly-supervised text classification allows users to build text classifiers by providing a handful of seed words, and it can often achieve performance that is a few percent lower than a fully-supervised model trained using thousands of labeled documents. However, the choice of seed words has a significant impact on weakly-supervised text classification's performance. The seed words used in previous work are neither trivial to come up, nor do we have access to labeled validation datasets to measure the performance and fine-tune the seed words in real-life applications. It casts doubts on the effectiveness of weakly-supervised text classification methods on new classification tasks.

In this work, we proposed a novel framework consisting of firstly using statistical methods to mine keywords associated with the category name, then using unsupervised Bayesian error estimation to directly estimate the impact of each seed word and compose the "optimal" set of seed words. The framework automatically composes seed words that yielded comparable performance as expert-curated keywords and outperformed baselines using only the category name as the seed word or using an unfiltered list of seed words.

We are actively working on extending the framework to multi-class classification by using one-vs.-rest or pair-wise classification. We are also investigating methods to incorporate the documents' similarity in the error estimation process to make the estimated error rate more accurate.

References

Jaime Carbonell and Jade Goldstein. 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*, pages 335–336.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 2, pages 830–835.

Nontawat Charoenphakdee, Jongyeong Lee, Yiping Jin, Dittaya Wanvarie, and Masashi Sugiyama. 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)*, pages 3984–3993.

Pinar Donmez, Guy Lebanon, and Krishnakumar Balasubramanian. 2010. Unsupervised supervised learning i: Estimating classification and regression errors without labels. *Journal of Machine Learning Research*, 11(4).

Gregory Druck, Gideon Mann, and Andrew McCallum. 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*, pages 595–602. ACM.

Noemie Elhadad and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Proceedings of Biological, Translational, and Clinical Language Processing*, pages 49–56, Prague, Czech Republic. Association for Computational Linguistics.

Evgeniy Gabrilovich, Shaul Markovitch, et al. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, volume 7, pages 1606–1611.

Zhen Hai, Kuiyu Chang, and Gao Cong. 2012. One seed to find them all: mining opinion features via association. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 255–264.

Minqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. In *Proceedings of the 19th National Conference on Artifical Intelligence*, pages 755–760.

Ariel Jaffe, Boaz Nadler, and Yuval Kluger. 2015. Estimating the accuracies of multiple classifiers without labeled data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 407–415.

Yiping Jin, Dittaya Wanvarie, and Phu Le. 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)*, volume 1, pages 545–554.

Yiping Jin, Dittaya Wanvarie, and Phu TV Le. 2020. Learning from noisy out-of-domain corpus using dataless classification. *Natural Language Engineering*, 1(1):1–35.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Daniel Kottke, Jim Schellinger, Denis Huseljic, and Bernhard Sick. 2019. Limitations of assessing active learning performance at runtime. *arXiv preprint arXiv:1901.10338*.

Chenliang Li, Shiqian Chen, Jian Xing, Aixin Sun, and Zongyang Ma. 2018. Seed-guided topic model for document filtering and classification. *ACM Transactions on Information Systems (TOIS)*, 37(1):1–37.

Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S Yu. 2004. Text classification by labeling words. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 4, pages 425–430.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 983–992.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6826–6833.

Emmanouil Antonios Platanios, Avrim Blum, and Tom Mitchell. 2014. Estimating accuracy from unlabeled data. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 682–691, Arlington, Virginia, USA. AUAI Press.

Emmanouil Antonios Platanios, Avinava Dubey, and Tom Mitchell. 2016. Estimating accuracy from unlabeled data: A bayesian approach. In *Proceedings of the International Conference on Machine Learning*, pages 1416–1425.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 105–112.

Burr Settles. 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*, pages 1467–1478. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.