SEMI-SUPERVISED TEXT CLASSIFICATION

A PREPRINT

Chengyu Dong
Computer Science & Engineering
cdong@eng.ucsd.edu

April 29, 2019

ABSTRACT

In this assignment we perform the sentiment analysis using supervised learning. Multiple feature engineering techniques are attempted and make prominent improvement. To utilize the underlying semantic information in the large unlabeled corpus, two types of semi-supervised strategies are attempted, giving close or slightly inferior performance compared to the supervised learning. Specific reasons are explored through detailed analyses.

***K*eywords** Sentiment analysis · semi-supervised learning

1 Introduction

In this assignment we will perform sentiment classification, through supervised and semi-supervised methods. The dataset consists of a labeled corpus of user reviews (about 5000 records), each annotated positive or negative, and also a large corpus of reviews that has no label (about 100,000 records). The task is to predict the sentiment of a review, based on a supervised model built on the labeled corpus, and with the help of unlabeled corpus further. To focus on the preprocessing and feature engineering part, we will use only simple logistic regression in this assignment.

2 Supervised: Improve the Basic Classifier

One simplest model to perform text classification is to treat each review as the bag of words, and thus can be represented by the occurrence of each word in this review. With a simple logistic regression classifier, this model achieved an accuracy of about 0.9821 on the training set and about 0.7773 on the hold-out set, which will be counted as a baseline for the following analysis.

We now try to design better features for classification. One thing that we can always try is to add more features. In this task we can use ngrams to enlarge the vocabulary. This will work because it will capture the relationships between words other than the simple occurrences of words themself. Using a combination of unigram, bigram and trigram, the accuracy of the model on the hold-out set increases to 0.7795, which is a minor improvement.

We can also do better than simply counting the occurrence of words in each review. One commonly used method is Tf-idf term weighting. It weights the occurrence of a word by its inverse document-frequency, which is defined as

$$\mathrm{idf}(t) = \log \frac{n}{1 + \mathrm{df}(t)},$$

where n is the total number of reviews, and $\mathrm{df}(t)$ is the number of the reviews that contain word t. Hence frequently used words like "the" and "a" will be weighted small since they carry little meaning information, while words like "excellent" and "awesome" will be weighted large because they rarely appear and carry intense positive attitudes. Using Tf-idf term weighting the accuracy of the model increases to 0.7838.

We can also focus on the tokenization part. The default regular expression pattern to identify a token is "\b\w\w+\b", which will capture "can't" as "can" and ignore the negation part. This will obviously impair the performance because

Table 1: Accuracies of the model achieved by several strategies. The performance in each row is achieved by the combinations of all the strategies above it. The last row corresponds the performance of a semi-supervised strategy, which will be discussed in Sect. 4

	Accuray	
Strategies	Train	Dev
Baseline	0.9821	0.7773
Ngram(1-3)	0.9998	0.7795
Tf-idf weighting	1.0	0.7838
Improved tokenizer	1.0	0.8188
Lemmatization & Spelling correction	1.0	0.8275
Word embedding & Similarity	1.0	0.8297

"can't" and "can" express totally different attitude but are mixed here. We can use a better tokenzier to capture "can't" as a combination of "ca" and "n't". This creates an unidentified word "ca" but "n't" will benefit the sentiment classification as a negative term. In practice we will use the built-in method *word_tokenize* offered by the natural language processing library *nltk*. The accuracy of the model is thereby increased to 0.8188, which is a huge gain.

One of the things that is crucial in text classification is to represent the text using built feature as far as possible. In English grammar we have the inflection, which often changes the ending of a word, for example the plural form of a noun. This causes a problem that some words in the test data will not be seen in the train data due to the change of forms, but they actually carry the same meanings. We can address this by lemmatization, which restore a word into its basic form. We will only lemmatize nouns in the corpus. Lemmatization of verbs or adjectives impairs the performance, due to unclear reasons. Another noticeable problem is spelling mistake. For example, in the unlabeled corpus we can find quite a few reviewers misspell "awesome" as "awsome". Sometimes the mistake is intentional. For example "gooood" is an intense way to express "good". These misspelled words will not be captured by the model simply because they are not contained in the vocabulary, but actually carry prominent emotional orientations. Proper spelling correction will thus benefit the classification, and by means of this technique we improve the model accuracy on the hold-out set to 0.8275, which is also a huge gain.

In Table. 1 we list all the successful strategies we implemented and their corresponding performance on the train and hold-out set. During the attempts we didn't change the hyper-parameters of the classifier for comparison. And on the basis of all these strategies we now fine-tune the hyper-parameters. But it turns out the original parameters of the logistic regression are already the best, namely the regularization coefficient is 1, the solver is "lbfgs", and the number of maximum iterations is 10000.

3 Semi-supervised: Exploiting the Unlabeled Data

We now try to utilize the unlabeled data to contribute to the classification, which is often call semi-supervised learning. We will use a simple strategy, that is to iteratively expand the labeled data with the most confident predictions given by the classifier on the unlabeled data, until there is no further prediction beyond the confidence threshold. This strategy can be described by the following procedure. Note that given a relatively low confidence threshold, the predicting confidence on all of the unlabeled data could be higher than the confidence level, under which circumstance all the

unlabeled corpus will be added to the training set. Thus the stopping criterion has another natural constraint, namely the size of the unlabeled corpus.

Algorithm 1: Iterative expansion

Input: Labled dataset D_l; unlabeled dataset D_u
Input: Confidence determination function $\mathcal{F}(D_l)$; Confidence threshold C_{th}
Output: Expanded labeled dataset $\hat{D}_l$

$\hat{D}_l \leftarrow D_l$;
$\hat{D}_u \leftarrow D_u$;
while $\max \mathcal{F}(\hat{D}_u) > C_{th}$ **do**
  $\theta \leftarrow \text{Train}(\hat{D}_l)$;
  $D_u^\dagger \leftarrow \text{Predict}(\hat{D}_u, \theta)$;
  $D_l^\dagger \leftarrow \{\hat{d}_l\}$ where $\mathcal{F}(\hat{d}_l) > C_{th}$ and $\hat{d}_l \in D_u^\dagger$;
  $\hat{D}_l \leftarrow \hat{D}_l + D_l^\dagger$;
  $\hat{D}_u \leftarrow \hat{D}_u - D_l^\dagger$;
end

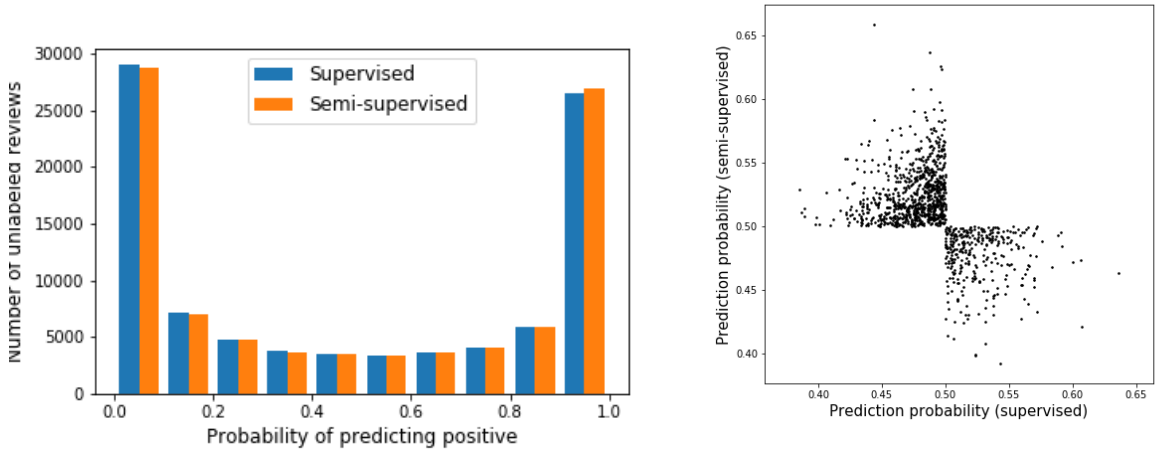
In practice, we will simply use the prediction probability given by the logistic classifier as confidence, and the confidence threshold will be set to 0.9999. After 3 loops, 2,367 examples in the unlabeled corpus are added to the labeled set, where 1,687 are with positive labels. The new model contains about 30% more features than the previous one. The performance of this semi-supervised model is slightly inferior to the above supervised model, with an accuracy of 0.8122 on the hold-out set. The statistics in each iteration are shown in Table. 2. We can clearly see that as more data are added to the training set in each iteration, the model performs worse. This is quite against our initial motivation.

With a detailed error analysis, we can figure out the reason for the inferior performance of our semi-supervised model. Compared to the predictions of supervised model, this model gives 9 different predictions on the hold-out set, among which there is only one correct prediction in terms of the ground-truth label. In other words, the semi-supervised model makes one more correct prediction at the cost of 8 mistakes. Interestingly, among the 9 different predictions, 8 are positive, which means that the semi-supervised model becomes more sensitive to positive reviews compared to supervised models, thus goes to extreme in this single direction. This tendency can be revealed more clearly by Fig. 1a, which shows the histograms of the probabilities of predicting positive for both models. The abundance of unlabeled corpus allows us to distinguish the tendency, that the model now predict more positive labels and less negative labels than before at each probability interval. We can also see in this graph that most of the data are easy, and the two models agree and both give high confidence. But in fact almost all mistakes happen on difficult examples, namely where the prediction probabilities are close to 0.5. This is shown in Fig. 1b, which compare the probabilities of predicting positive given by the models only on the unlabeled data they disagree. We can again clearly see that there are much more data predicted from negative to positive than the opposite way.

We now show some specific examples that the semi-supervised model overturns the supervised one and make mistakes. Some reviews that the two models disagree the most in terms of probability are as follows. These two reviews are both negative in obvious, and the supervised model gets them right. However, while doing semi-supervised adaption, the classifier trained by the expanded labeled data fits an intercept that is slightly larger, implying an overall positive tendency and leading to the misclassification of this example. Though the change of intercept is small, it will affect the ambiguous examples that lie close to the decision boundary heavily.

- `This Sonic had good food (for a Sonic) but service was very slow.Be advised that they use fake whipped cream as a topping on their shakes and (presumably) ice cream.`
- `Wow, what a disappointment!  The boyfriend and I were celebrating out anniversary by staying at the Pointe and decided to try this place for dinner on a Saturday night.  At`

So after all what is the underlying reason for this overall positive tendency? Recall that in the self-training procedure most of the examples (>70%) added to the training set are labeled as positive. And since the original training set only consists of 4,582 examples, this bias of expanded labels can affect the decision boundary of the logistic regression heavily, meaning that it could move toward the positive side and thus misclassifies examples on the boundary. And this effect will be enhanced in each iteration. A drift of decision boundary toward positive side will make it predict



(a) Histograms of the probabilities of predicting positive for supervised and semi-supervised models

(b) The probabilities of predicting positive by supervised model versus that by semi-supervised model, only on the unlabeled data that their predictions disagree.

Figure 1: Comparison of predictions of supervised and semi-supervised models on the unlabeled corpus

Table 2: Number of data added to the training set with positive and negative labels in each iteration, as well as the resulted accuracy on the hold-out set.

Iteration	# positive labels added	# negative labels added	Accuracy
0	-	-	0.8275
1	1664	660	0.8144
2	20	19	0.8122
3	3	1	0.8122

more positive labels confidently, and expansion of these "seemingly" confident examples to training will deteriorate the biased drift further. However, if we use a discriminative classifier such as Support Vector Machine (SVM), this problem may be able to solved properly.

Now we compare the two models in terms of features. Since we are using the tf-idf weighting, the importance of the features are both determined by the coefficients fitted by the classifier and the inverse document-frequency that is inherent to the entire dataset. First we list some words whose idfs changes the most and also exhibits differentiable orientations, which is shown in Table. 3. We can clearly see that most of the features whose idf changes a lot are positive terms, simply because most of the data added to the labeled corpus are positive, which then heavily affect the overall weighting of positive terms. In Table. 4 we list some words whose regression coefficient changes the most. We can see that these are still be dominated by positive terms. That the coefficients of some trivial words like and and is change abruptly indicates that self-training propagates useless information which may be harmful to the decision. One noticeable thing is that the absolute values of the coefficients of almost all the features decrease, this is the offset effect of the overall positive tendency, reflected in the increase of regression intercept.

Finally we will examine how our semi-supervised model performs when the hyper-parameters vary. We are interested in the behavior of the model as the number of the labeled data that are offered to training. Fig. 2a shows the accuracy that the model achieved on the hold-out set versus the fraction of the labeled data offered, as well as the number of unlabeled data that are added to the training set after self-training. It is natural that as less labeled data are offered, the performance becomes worse. We also annotate the result of supervised learning on the graph for comparison.

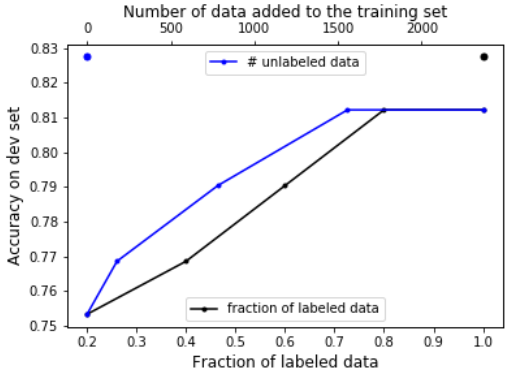
In Fig. 2b we further show the accuracy the semi-supervised model on the hold-out set as the confidence threshold varies. Higher confidence threshold means less unlabeled data will be added to training. As confidence threshold increase, the performance of the model keeps improving, and converges to the supervised result where the confidence threshold is equivalent to 1.

Table 3: Top 10 meaningful features whose inverse document-frequencies change the most after self-training and their potential emotional orientation, only in consideration of unigrams for simplicity.

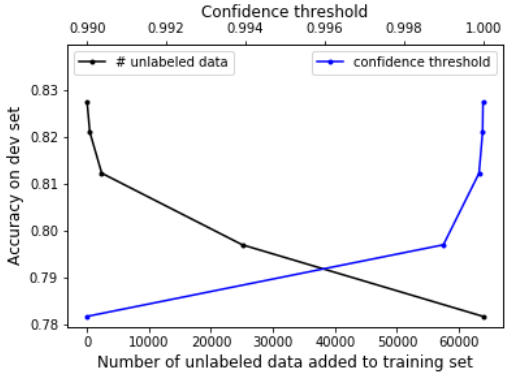
	IDF (Supervised)	IDF (Semi-supervised)	Orientation
delicious	4.5549	3.6853	positive
refund	8.7370	7.9006	negative
amazing	4.3181	3.4991	positive
favourite	4.8451	4.0535	positive
personable	7.6383	6.8508	positive
homemade	7.3507	6.5884	positive
friendly	4.1120	3.3850	positive
scratch	8.7370	8.0547	negative
refused	8.7370	8.0547	negative
yummy	6.2112	5.5698	positive

Table 4: Top 10 features whose regression coefficients change the most after self-training, only in consideration of considera-unigrams for simplicity.

	coefficient (Supervised)	coefficient (Semi-supervised)
!	0.4301	0.3780
ever	-0.1018	-0.0688
great	0.5419	0.5102
and	0.2320	0.2069
is	0.0946	0.0701
love	0.3056	0.2821
best	0.3506	0.3311
are	0.0323	0.0130
worst	-0.4646	-0.4464
very	0.1303	0.1124



(a) The accuracy of the semi-supervised model versus the fraction of labeled data used to train, as well as the number of unlabeled data that are added to the training set as the number of labeled data varies. The large dot indicates the supervised result.



(b) The accuracy of the semi-supervised model versus the confidence threshold, as well as the number of unlabeled data that are added to the training set as confidence threshold varies. The case when threshold equals to 1 is the same as supervised learning.

4 Semi-supervised: Designing better features

The sparsity of the vocabulary in the training data means that a number of tokens in the unlabeled data will be neglected during transformation. If we can find a way to alternatively represent these unseen tokens, the unlabeled data can then be better fitted into the model. In this section we will try to test a simple strategy, which first trains an embedding for each token in the unlabeled corpus using the corpus itself, then replace the unseen token with its most similar available token in the model vocabulary in terms of cosine similarities between embeddings.

We will utilize the famous Continuous Bag of Words (CBOW) model, which tries to predict the center word based on its surrounding words in the corpus. Hence it leverages the semantics and will build similar embeddings if the contexts are similar. We will set the word embedding length to be 100, and the window size will be set to 5, which means the center word will be predicted using both 5 words ahead and behind it in a sentence. After the CBOW model is trained and word embeddings are built, we can now modify the feature engineering part in the supervised model. Specifically, during prediction, if any token in the unlabeled corpus is not contained in the vocabulary built from the training set, we will check a list of most similar 20 tokens from top to bottom in the corpus in terms of cosine similarity between word embeddings, and replace this unseen token with the first similar token that is contained in the vocabulary. By means of this method, we further improve the prediction accuracy on the hold-set to 0.8297, as listed in Table. 1.

Now we look into the specific examples that this strategy causes different. Compared to the supervised model, this semi-supervised strategy disagrees on 9 reviews, among which it gets 5 right, while the supervised model gets 4 right. So essentially it corrects 5 predictions, but makes 4 more mistakes. The first review in the following is an example

the semi-supervised strategy works. It is in German, which makes sense because the similarity information will be captured by the word embedding model trained on the unseen data, especially for a small part of the corpus that are represented by totally different tokens, but carry semantic information inherently. These information summarized by the supervised model from a very small subset will then be effectively propagated to other words in this language. In fact among the 5 correct predictions that this semi-supervised strategy achieved, 3 are in foreign languages, either in German or French. The second review in the following is an example the strategy makes a mistake, while is predicted correctly by the original supervised model. In fact in this review the token "41,300" is not contained in the vocabulary, thus it is replaced by its most similar word "excellent" in the CBOW model. Presumably the unlabeled set contains another review says about the "car" is "excellent", thus the information is propagated to this review, but it makes no sense. Therefore the word embedding information could also be misleading.

- `Seeeeeeeeeeehr wunderbares Geschaeft fuer Menschen mit der Liebe zum Espresso.Herr Koeberl verkauft nicht nur Espressomaschinen, nein, man kann auch einen herausragend leckeren Espresso (oder auch 2) bei ihm trinken.Er fuehrt`
- `Took my car in to get a quote on the brakes, my car is a 2011 with only 41,300 miles on it.  They wanted to replace all the pads`