

# Credibility Adjusted Term Frequency: A Supervised Term Weighting Scheme for Sentiment Analysis and Text Classification

Yoon Kim

New York University  
yhk255@nyu.edu

Owen Zhang

zhonghua.zhang2006@gmail.com

## Abstract

We provide a simple but novel supervised weighting scheme for adjusting term frequency in *tf-idf* for sentiment analysis and text classification. We compare our method to baseline weighting schemes and find that it outperforms them on multiple benchmarks. The method is robust and works well on both snippets and longer documents.

## 1 Introduction

Baseline discriminative methods for text classification usually involve training a linear classifier over bag-of-words (BoW) representations of documents. In BoW representations (also known as Vector Space Models), a document is represented as a vector where each entry is a count (or binary count) of tokens that occurred in the document. Given that some tokens are more informative than others, a common technique is to apply a weighting scheme to give more weight to discriminative tokens and less weight to non-discriminative ones. Term frequency-inverse document frequency (*tf-idf*) (Salton and McGill, 1983) is an unsupervised weighting technique that is commonly employed. In *tf-idf*, each token  $i$  in document  $d$  is assigned the following weight,

$$w_{i,d} = tf_{i,d} \cdot \log \frac{N}{df_i} \quad (1)$$

where  $tf_{i,d}$  is the number of times token  $i$  occurred in document  $d$ ,  $N$  is the number of documents in the corpus, and  $df_i$  is the number of documents in which token  $i$  occurred.

Many supervised and unsupervised variants of *tf-idf* exist (Debole and Sebastiani (2003); Martineau and Finin (2009); Wang and Zhang (2013)). The purpose of this paper is not to perform an exhaustive comparison of existing weighting

schemes, and hence we do not list them here. Interested readers are directed to Paltoglou and Thelwall (2010) and Deng et al. (2014) for comprehensive reviews of the different schemes.

In the present work, we propose a simple but novel supervised method to adjust the term frequency portion in *tf-idf* by assigning a credibility adjusted score to each token. We find that it outperforms the traditional unsupervised *tf-idf* weighting scheme on multiple benchmarks. The benchmarks include both snippets and longer documents. We also compare our method against Wang and Manning (2012)’s Naive-Bayes Support Vector Machine (NBSVM), which has achieved state-of-the-art results (or close to it) on many datasets, and find that it performs competitively against NBSVM. We additionally find that the traditional *tf-idf* performs competitively against other, more sophisticated methods when used with the right scaling and normalization parameters.

## 2 The Method

Consider a binary classification task. Let  $C_{i,k}$  be the count of token  $i$  in class  $k$ , with  $k \in \{-1, 1\}$ . Denote  $C_i$  to be the count of token  $i$  over both classes, and  $y^{(d)}$  to be the class of document  $d$ . For each occurrence of token  $i$  in the training set, we calculate the following,

$$s_i^{(j)} = \begin{cases} \frac{C_{i,1}}{C_i} & , \text{ if } y^{(d)} = 1 \\ \frac{C_{i,-1}}{C_i} & , \text{ if } y^{(d)} = -1 \end{cases} \quad (2)$$

Here,  $j$  is the  $j$ -th occurrence of token  $i$ . Since there are  $C_i$  such occurrences,  $j$  indexes from 1 to  $C_i$ . We assign a score to token  $i$  by,

$$\hat{s}_i = \frac{1}{C_i} \sum_{j=1}^{C_i} s_i^{(j)} \quad (3)$$

Intuitively,  $\hat{s}_i$  is the average likelihood of making the correct classification given token  $i$ ’s occurrence in the document, if  $i$  was the only token in

the document. In a binary classification case, this reduces to,

$$\hat{s}_i = \frac{C_{i,1}^2 + C_{i,-1}^2}{C_i^2} \quad (4)$$

Note that by construction, the support of  $\hat{s}_i$  is  $[0.5, 1]$ .

## 2.1 Credibility Adjustment

Suppose  $\hat{s}_i = \hat{s}_j = 0.75$  for two different tokens  $i$  and  $j$ , but  $C_i = 5$  and  $C_j = 100$ . Intuition suggests that  $\hat{s}_j$  is a more credible score than  $\hat{s}_i$ , and that  $\hat{s}_i$  should be shrunk towards the population mean. Let  $\hat{s}$  be the (weighted) population mean. That is,

$$\hat{s} = \sum_i \frac{C_i \cdot \hat{s}_i}{C} \quad (5)$$

where  $C$  is the count of all tokens in the corpus. We define *credibility adjusted score* for token  $i$  to be,

$$\bar{s}_i = \frac{C_{i,1}^2 + C_{i,-1}^2 + \hat{s} \cdot \gamma}{C_i^2 + \gamma} \quad (6)$$

where  $\gamma$  is an additive smoothing parameter. If  $C_{i,k}$ 's are small, then  $\bar{s}_i \approx \hat{s}$  (otherwise,  $\bar{s}_i \approx \hat{s}_i$ ). This is a form of Buhlmann credibility adjustment from the actuarial literature (Buhlmann and Gisler, 2005). We subsequently define  $\bar{tf}$ , the *credibility adjusted term frequency*, to be,

$$\bar{tf}_{i,d} = (0.5 + \bar{s}_i) \cdot tf_{i,d} \quad (7)$$

and  $tf$  is replaced with  $\bar{tf}$ . That is,

$$w_{i,d} = \bar{tf}_{i,d} \cdot \log \frac{N}{df_i} \quad (8)$$

We refer to above as *cred-tf-idf* hereafter.

## 2.2 Sublinear Scaling

It is common practice to apply sublinear scaling to  $tf$ . A word occurring (say) ten times more in a document is unlikely to be ten times as important. Paltoglou and Thelwall (2010) confirm that sublinear scaling of term frequency results in significant improvements in various text classification tasks. We employ logarithmic scaling, where  $tf$  is replaced with  $\log(tf) + 1$ . For our method,  $\bar{tf}$  is simply replaced with  $\log(\bar{tf}) + 1$ . We found virtually no difference in performance between log scaling and other sublinear scaling methods (such as augmented scaling, where  $tf$  is replaced with  $0.5 + \frac{0.5+tf}{\max tf}$ ).

## 2.3 Normalization

Using normalized features resulted in substantial improvements in performance versus using un-normalized features. We thus use  $\hat{\mathbf{x}}^{(d)} = \mathbf{x}^{(d)} / \|\mathbf{x}^{(d)}\|_2$  in the SVM, where  $\mathbf{x}^{(d)}$  is the feature vector obtained from *cred-tf-idf* weights for document  $d$ .

## 2.4 Naive-Bayes SVM (NBSVM)

Wang and Manning (2012) achieve excellent (sometimes state-of-the-art) results on many benchmarks using binary Naive Bayes (NB) log-count ratios as features in an SVM. In their framework,

$$w_{i,d} = \mathbf{1}\{tf_{i,d}\} \log \frac{(df_{i,1} + \alpha) / \sum_i (df_{i,1} + \alpha)}{(df_{i,-1} + \alpha) / \sum_i (df_{i,-1} + \alpha)} \quad (9)$$

where  $df_{i,k}$  is the number of documents that contain token  $i$  in class  $k$ ,  $\alpha$  is a smoothing parameter, and  $\mathbf{1}\{\cdot\}$  is the indicator function equal to one if  $tf_{i,d} > 0$  and zero otherwise. As an additional benchmark, we implement NBSVM with  $\alpha = 1.0$  and compare against our results.<sup>1</sup>

## 3 Datasets and Experimental Setup

We test our method on both long and short text classification tasks, all of which were used to establish baselines in Wang and Manning (2012). Table 1 has summary statistics of the datasets. The snippet datasets are:

- **PL-sh**: Short movie reviews with one sentence per review. Classification involves detecting whether a review is positive or negative. (Pang and Lee, 2005).<sup>2</sup>
- **PL-sub**: Dataset with short subjective movie reviews and objective plot summaries. Classification task is to detect whether the sentence is objective or subjective. (Pang and Lee, 2004).

And the longer document datasets are:

<sup>1</sup>Wang and Manning (2012) use the same  $\alpha$  but they differ from our NBSVM in two ways. One, they use  $l_2$  hinge loss (as opposed to  $l_1$  loss in this paper). Two, they interpolate NBSVM weights with Multivariable Naive Bayes (MNB) weights to get the final weight vector. Further, their tokenization is slightly different. Hence our NBSVM results are not directly comparable. We list their results in table 2.

<sup>2</sup><https://www.cs.cornell.edu/people/pabo/movie-review-data/>. All the PL datasets are available here.

Dataset	Length	Pos	Neg	Test
PL-sh	21	5331	5331	CV
PL-sub	24	5000	5000	CV
PL-2k	746	1000	1000	CV
IMDB	231	12.5k	12.5k	25k
AthR	355	480	377	570
XGraph	276	584	593	784

Table 1: Summary statistics for the datasets. Length is the average number of unigram tokens (including punctuation) per document. Pos/Neg is the number of positive/negative documents in the training set. Test is the number of documents in the test set (CV means that there is no separate test set for this dataset and thus a 10-fold cross-validation was used to calculate errors).

- **PL-2k**: 2000 full-length movie reviews that has become the de facto benchmark for sentiment analysis (Pang and Lee, 2004).
- **IMDB**: 50k full-length movie reviews (25k training, 25k test), from IMDB (Maas et al., 2011).<sup>3</sup>
- **AthR, XGraph**: The 20-Newsgroup dataset, 2nd version with headers removed.<sup>4</sup> Classification task is to classify which topic a document belongs to. AthR: alt.atheism vs religion.misc, XGraph: comp.windows.x vs comp.graphics.

### 3.1 Support Vector Machine (SVM)

For each document, we construct the feature vector  $\mathbf{x}^{(d)}$  using weights obtained from *cred-tf-idf* with log scaling and  $l_2$  normalization. For *cred-tf-idf*,  $\gamma$  is set to 1.0. NBSVM and *tf-idf* (also with log scaling and  $l_2$  normalization) are used to establish baselines. Prediction for a test document is given by

$$y^{(d)} = \text{sign}(\mathbf{w}^T \mathbf{x}^{(d)} + b) \quad (10)$$

In all experiments, we use a Support Vector Machine (SVM) with a linear kernel and penalty parameter of  $C = 1.0$ . For the SVM,  $\mathbf{w}$ ,  $b$  are obtained by minimizing,

$$\mathbf{w}^T \mathbf{w} + C \sum_{d=1}^N \max(0, 1 - y^{(d)}(\mathbf{w}^T \mathbf{x}^{(d)} + b)) \quad (11)$$

using the LIBLINEAR library (Fan et al., 2008).

<sup>3</sup><http://ai.stanford.edu/amaas/data/sentiment/index.html>

<sup>4</sup><http://people.csail.mit.edu/jrennie/20Newsgroups>

### 3.2 Tokenization

We lower-case all words but do not perform any stemming or lemmatization. We restrict the vocabulary to all tokens that occurred at least twice in the training set.

## 4 Results and Discussion

For PL datasets, there are no separate test sets and hence we use 10-fold cross validation (as do other published results) to estimate errors. The standard train-test splits are used on IMDB and Newsgroup datasets.

### 4.1 *cred-tf-idf* outperforms *tf-idf*

Table 2 has the comparison of results for the different datasets. Our method outperforms the traditional *tf-idf* on all benchmarks for both unigrams and bigrams. While some of the differences in performance are significant at the 0.05 level (e.g. IMDB), some are not (e.g. PL-2k). The Wilcoxon signed ranks test is a non-parametric test that is often used in cases where two classifiers are compared over multiple datasets (Demsar, 2006). The Wilcoxon signed ranks test indicates that the overall outperformance is significant at the  $<0.01$  level.

### 4.2 NBSVM outperforms *cred-tf-idf*

*cred-tf-idf* did not outperform Wang and Manning (2012)’s NBSVM (Wilcoxon signed ranks test  $p$ -value = 0.1). But it did outperform our own implementation of NBSVM, implying that the extra modifications by Wang and Manning (2012) (i.e. using squared hinge loss in the SVM and interpolating between NBSVM and MNB weights) are important contributions of their methodology. This was especially true in the case of shorter documents, where our uninterpolated NBSVM performed significantly worse than their interpolated NBSVM.

### 4.3 *tf-idf* still performs well

We find that *tf-idf* still performs remarkably well with the right scaling and normalization parameters. Indeed, the traditional *tf-idf* outperformed many of the more sophisticated methods that employ distributed representations (Maas et al. (2011); Socher et al. (2011)) or other weighting schemes (Martineau and Finin (2009); Deng et al. (2014)).

	Method	PL-sh	PL-sub	PL-2k	IMDB	AthR	XGraph
Our results	tf-idf-uni	77.1	91.5	88.1	88.6	85.8	88.4
	tf-idf-bi	78.0	92.3	89.2	90.9	86.5	88.0
	cred-tfidf-uni	77.5	91.8	88.7	88.8	86.5	89.8
	cred-tfidf-bi	78.6	<b>92.8</b>	<b>89.7</b>	<b>91.3</b>	<b>87.4</b>	88.9
	NBSVM-uni	75.5	89.9	87.0	85.9	86.7	88.5
	NBSVM-bi	76.0	90.5	89.5	90.5	86.7	88.1
Wang & Manning	MNB-uni	77.9	92.6	83.5	83.6	85.0	90.0
	MNB-bi	<b>79.0</b>	<b>93.6</b>	85.9	86.6	85.1	<b>91.2</b>
	NBSVM-uni	78.1	92.4	87.8	88.3	<b>87.9</b>	<b>91.2</b>
	NBSVM-bi	<b>79.4</b>	<b>93.2</b>	<b>89.5</b>	<b>91.2</b>	<b>87.7</b>	<b>90.7</b>
Other results	Appr. Tax.*	-	-	90.2	-	-	-
	Str. SVM*	-	-	92.4	-	-	-
	aug-tf-mi	-	-	87.8	88.0	-	-
	Disc. Conn.	-	-	-	<b>91.4</b>	-	-
	Word Vec.*	-	88.6	88.9	88.9	-	-
	LLR	-	-	<b>90.4</b>	-	-	-
	RAE	77.7	-	-	-	-	-
	MV-RNN	<b>79.0</b>	-	-	-	-	-

Table 2: Results of our method (*cred-tfidf*) against baselines (*tf-idf*, NBSVM), using unigrams and bigrams. *cred-tfidf* and *tf-idf* both use log scaling and  $l_2$  normalization. Best results (that do not use external sources) are underlined, while top three are in bold. Rows 7-11 are MNB and NBSVM results from Wang and Manning (2012). Our NBSVM results are not directly comparable to theirs (see footnote 1). Methods with \* use external data or software. **Appr. Tax.**: Uses appraisal taxonomies from *WordNet* (Whitelaw et al., 2005). **Str. SVM**: Uses *OpinionFinder* to find objective versus subjective parts of the review (Yessenalina et al., 2010). **aug-tf-mi**: Uses augmented term-frequency with mutual information gain (Deng et al., 2014). **Disc. Conn.**: Uses discourse connectors to generate additional features (Trivedi and Eisenstein, 2013). **Word Vec.**: Learns sentiment-specific word vectors to use as features combined with BoW features (Maas et al., 2011). **LLR**: Uses log-likelihood ratio on features to select features (Aue and Gamon, 2005). **RAE**: Recursive autoencoders (Socher et al., 2011). **MV-RNN**: Matrix-Vector Recursive Neural Networks (Socher et al., 2012).

## 5 Conclusions and Future Work

In this paper we presented a novel supervised weighting scheme, which we call *credibility adjusted term frequency*, to perform sentiment analysis and text classification. Our method outperforms the traditional *tf-idf* weighting scheme on multiple benchmarks, which include both snippets and longer documents. We also showed that *tf-idf* is competitive against other state-of-the-art methods with the right scaling and normalization parameters.

From a performance standpoint, it would be interesting to see if our method is able to achieve even better results on the above tasks with proper tuning of the  $\gamma$  parameter. Relatedly, our method could potentially be combined with other supervised variants of *tf-idf*, either directly or through ensembling, to improve performance further.

## References

- A. Aue, M. Gamon. 2005. Customizing sentiment classifiers to new domains: A case study. *Proceedings of the International Conference on Recent Advances in NLP, 2011*.
- H. Buhlmann, A. Gisler. 2005. A Course in Credibility Theory and its Applications *Springer-Verlag, Berlin*.
- F. Debole, F. Sebastiani. 2003. Supervised Term Weighting for Automated Text Categorization *Proceedings of the 2003 ACM symposium on Applied Computing, 784–788*.
- J. Demsar. 2006. Statistical Comparison of classifiers over multiple data sets. *Journal of Machine Learning Research, 7:1-30, 2006*.
- Z. Deng, K. Luo, H. Yu. 2014. A study of supervised term weighting scheme for sentiment analysis *Ex-*

- pert Systems with Applications. Volume 41, Issue 7, 3506–3513.*
- R. Fan, K. Chang, J. Hsieh, X. Wang, C. Lin. 2008. LI-BLINEAR: A library for large linear classification. *Journal of Machine Learning Research, 9:1871–1874, June.*
- A. Maas, R. Daly, P. Pham, D. Huang, A. Ng, C. Potts. 2011. Learning Word Vectors for Sentiment Analysis. *In Proceedings of ACL 2011.*
- J. Martineau, T. Finin. 2009. Delta TFIDF: An Improved Feature Space for Sentiment Analysis. *Third AAAI International Conference on Weblogs and Social Media*
- G. Paltoglou, M. Thelwall. 2010. A study of Information Retrieval weighting schemes for sentiment analysis. *In Proceedings of ACL 2010.*
- B. Pang, L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *In Proceedings of ACL 2004.*
- B. Pang, L. Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *In Proceedings of ACL 2005.*
- R. Socher, J. Pennington, E. Huang, A. Ng, C. Manning. 2011. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. *In Proceedings of EMNLP 2011.*
- R. Socher, B. Huval, C. Manning, A. Ng. 2012. Semantic Compositionality through Recursive Matrix-Vector Spaces. *In Proceedings of EMNLP 2012.*
- R. Trivedi, J. Eisenstein. 2013. Discourse Connectors for Latent Subjectivity in Sentiment Analysis. *In Proceedings of NAACL 2011.*
- G. Salton, M. McGill. 1983. Introduction to Modern Information Retrieval. *McGraw-Hill.*
- S. Wang, C. Manning. 2012. Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. *In proceedings of ACL 2012.*
- D. Wang, H. Zhang. 2013. Inverse-Category-Frequency Based Supervised Term Weighting Schemes for Text Categorization. *Journal of Information Science and Engineering 29, 209–225.*
- C. Whitelaw, N. Garg, S. Argamon. 2005. Using appraisal taxonomies for sentiment analysis. *In Proceedings of CIKM 2005.*
- A. Yessenalina, Y. Yue, C. Cardie. 2010. Multi-level Structured Models for Document-level Sentiment Classification. *In Proceedings of ACL 2010.*