Enhancing Sentiment Analysis with Term Sentiment Entropy: Capturing Nuanced Sentiment in Text Classification

Suttipong Klongdee and Jatsada Singthongchai *

Posted Date: 22 March 2024

doi: 10.20944/preprints202403.1364.v1

Keywords: sentiment analysis; term weighting; text classification; TF-IDF; TFRF; natural language processing

Article

Enhancing Sentiment Analysis with Term Sentiment Entropy: Capturing Nuanced Sentiment in Text Classification

Suttipong Klongdee [1] and Jatsada Singthongchai [2,*]

[1] Department of Information Technology Faculty of Social Technology, Rajamangala University of Technology Tawan-ok, Chanthaburi Campus, 22210, Thailand; suttipong.rmutto@gmail.com
[2] Department of Computer Science and Information Technology, Faculty of Sciences and Health Technology, Kalasin University, 46000, Thailand
* Correspondence: jatsada.si@ksu.ac.th

Abstract: Sentiment analysis plays a crucial role in understanding customer feedback, guiding product development, and informing business decisions. This paper introduces term sentiment entropy (TSE), a novel weighting method that leverages the distribution of sentiment labels associated with words to enhance text classification accuracy. TSE complements traditional TF-IDF techniques by capturing the nuances of sentiment variation across different contexts. Experiments across diverse public datasets demonstrate TSE's potential to improve sentiment analysis performance, especially in capturing subtle sentiment shifts and adapting to specific domains. While computational cost and label quality pose challenges, TSE offers a promising avenue for refining sentiment analysis and opening new research frontiers.

Keywords: sentiment analysis; term weighting; text classification; TF-IDF; TFRF; natural language processing

1. Introduction

In today's digital word, customer feel free to post, review or feedback to provider over web board, mobile application or social media. The understanding what customer thinks and feels is paramount. There are many lesson learn that a drama message of some customer can explode and impact to a business. And no longer can businesses reply on traditional surveys.

In the other way, E-commerce is more popular because the improvement of information technology. It made the main communication way that client use, is messaging. The client sentiment was used to develop new product that closed to customer's expectations [1]. This is where sentiment analysis steps in, it is tools to extract emotions embedded within the message that sent in digital forms.

The insight information from sentiment analysis, business can Enhanced Customer Experience [2,6,8], Refined Marketing Strategies [3,9,10], Proactive Crisis Management [4,11], Data-Driven Product Development [5,39–41], Improved Employee Engagement [7]. Sentiment analysis is a game-changer for businesses in the digital age. In the other hands, sentiment analysis able to understand generalizable knowledge about public attitudes towards complex events like pandemics COVIC-19 [38]

In terms of human understanding often classify sentences according to their overall meaning. While each word has its own meaning. Words that can be interpreted on their own that correlate or opposite with some words and sentences can contain many groups of words that can combine and make it difference meanings. The meaning of the sentence depends on the word order and word group.

But computers cannot make analysis text directly. It need to convert text to array of number that represent to a word or group of words. To do that sentence will be separated into words and use term as representative. This method called tokenization or vectorization. The method can separate in 2 types. The 1st one is unsupervised technique that have some popular technique such as TF.IDF, TF-

 doi:10.20944/preprints202403.1364.v1

ISF, RTF-SISF, and TF-RF. And 2nd one is supervised technique such as "term weighting using class mutual information: RF-IDF-CMI" [1] was propose to find relationship between word and attention class. There is a result of RF-IDF-CMI that shows better accuracy than TF-IDF and TF-RF [1]. Marco [37] studied show preprocessing can significantly impact the performance of text classification models, even modern Transformers.

There are many traditional text classification algorithms that was propose for sentiment analysis such as Naive Bayes [1–4,9], Random forest [1], K-NN [1], SVM [2,4], Entropy [3], Lexicon-based [4,5,10], Latent Dirichlet Allocation (LDA) [6], Regression [9]. It also have number of specific algorithms that was used in each paper.

This paper propose new tokenization method called "sentence entropy" that apply term entropy weight instead of term frequency weight.

2. Materials and Methods

Term-frequency Invert-document-frequency "TF-IDF" is a popular technique used for feature extraction in sentiment analysis, helping to identify the most relevant words and phrases within a document that contribute to its sentiment. Benefits of this method is focusing on informative words, reducing noise, and simple. Limitation is ignores word order, ignores context and sensitive to pre-processing, stop word removal and stemming easily affect TF-IDF scores. [12–16]

TF-IDF can calculate by following formula.

$$tf\_idf_{ij} = tf_{ij} \cdot idf_i \tag{1}$$

$$tf_{ij} = \frac{t_{ij}}{T_j} \tag{2}$$

$$idf_i = \log \frac{D}{d_i} \tag{3}$$

When

t_{ij} : Count of term i in document j

T_j : Count of all term in document j

D : Count of document in dataset

d_i : Count of document that term i exist

Class mutual information "CMI" for positive sentiment builds from IDF weight weakness observation. The weakness is IDF calculate based on document frequency that specific term appearance. As show on Figure 1, IDF weight will be comparable between t1, t2, and t3. As well as t4, t5, and t6. However if considerate the bias of population in term of sentiment. t1 and t4 shows positive. t2 and t4 shows neutral. But t3 and t6 shows negative. To fix this weakness the researcher proposed a select class mutual information as a weight.

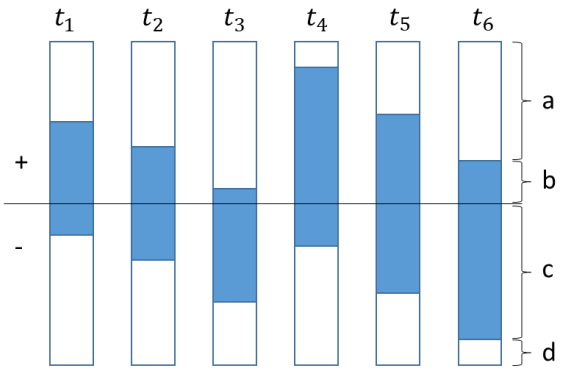

Figure 1. example sentiment distribution of specific term appearance on all document. a : number of positive document that term i doesn't exist. b : number of positive document that term i exist. c :

 doi:10.20944/preprints202403.1364.v1

number of negative document that term i exist. d : number of negative document that term i doesn't exist.

The mutual information is a concept in probability theory and information theory that quantifies the mutual dependence between two random variables. It is highly relate with Entropy. The entropy is another probability theory that shows the uncertainty of a single variable. They can calculate with following formula.

$$I_{(X;Y)} = \sum_{x\epsilon\Omega}\sum_{y\epsilon\Psi} p(x,y)\log\frac{p(x,y)}{p(x)\cdot p(y)} \tag{4}$$

$$H(X) = -\sum_{x\epsilon\Omega} p(x)\log p(x) \tag{5}$$

When

"$I_{(X;Y)}$" is probability of 2 decrease random valuable in all possible value "$x\epsilon\Omega$ and $y\epsilon\Psi$"

"$H(X)$" is probability of a random valuable in all possible value "$x\epsilon\Omega$"

CMI can calculate by following formula.

$$tf.idf.cmi_{ij} = tf_{ij}\cdot idf_i \cdot cmi_{iq} \tag{6}$$

$$cmi_{iq} = I_{(X=\{t_i\}:Y=\{c_q\})} \tag{7}$$

$$I_{(X=\{t_i\}:Y=(pos\})} = \frac{b}{N}\log\frac{\frac{b}{N}}{\left(\frac{b+c}{N}\right)\cdot\left(\frac{a+b}{N}\right)} \tag{8}$$

When

a : number of positive document that has no term i support

b : number of positive document that has some term i support

c : number of negative document that has some term i support

N : number of all document

This method shows better efficiency of document search. However, the researcher concern about the convert term frequency to binary before the calculation will impact running time and missing some information and a select class mutual information may bias if using imbalance sentiment word of each sentence. The recommendation is using mutual information from other method such as Fuzzy sets or Case based reasoning.

This paper propose Term Sentiment Entropy "TSE". Instead of simply focusing on how frequent a word is (TF) or how unique it is across documents (IDF), term sentiment entropy captures the distribution of sentiment labels associated with that word across different documents.

Imagine a word like "happy." It might appear frequently in positive reviews (high TF), but it might also occasionally appear in neutral or even negative contexts. Term sentiment entropy would quantify this variation in sentiment labels associated with "happy."

The specifics of calculating TSE is shows in following formula.

$$tf.idf.tse_{ij} = tf_{ij}\cdot idf_i \cdot tse_i \tag{9}$$

$$tse_i = \frac{1}{-\sum_{q\epsilon\text{classes}}\left[\frac{m_{iq}}{N}\log\left(\frac{m_{iq}}{N}\right)\right]} \tag{10}$$

When

m : number of document has q class that has term i support

N : number of all document

 doi:10.20944/preprints202403.1364.v1

High entropy would indicate that the word appears in documents with diverse sentiment labels (e.g., "happy" in reviews from happy, neutral, and disappointed customers). Low entropy would suggest a strong association with a specific sentiment (e.g., "gloomy" appearing mostly in negative reviews).

This entropy value can then be used as an additional weight for the word, potentially refining sentiment analysis in several ways:

- Improved accuracy: Words with high entropy might be less informative for specific sentiment classifications, so down weighting them could lead to more accurate results.
- Domain adaptation: Words with low entropy might be specific to a certain domain (e.g., "scam" in review sites), and incorporating their entropy could improve sentiment analysis in that domain.
- Identifying nuanced sentiment: Words with high entropy might be valuable for capturing subtle sentiment changes within texts, allowing for more advanced sentiment analysis.

Term sentiment entropy complements TF and IDF by adding a layer of sentiment-specific information:

- TF focuses on frequency: High TF words appear often, but they might not be sentiment-specific.
- IDF focuses on uniqueness: High IDF words are rare, but they might not be relevant for overall sentiment.
- Term sentiment entropy focuses on sentiment distribution: It provides a nuanced understanding of how a word's sentiment varies across different contexts.

By combining these metrics, sentiment analysis can potentially become more accurate, robust, and capable of capturing finer shades of sentiment within texts.

Comparing feature extraction calculation shows TF is the the feature that represent value of each term that depending document. While IDF, CMI, and TSE are represent value of each term across all document.

The IDF indicate to the term is unique for the document. That mean the small appearance term will return highly sensitive for the class of the document.

The CMI indicate to the term is high probability as a focus class, positive class as example, and high probability of appearance. This way possible to bias focus on a specific class and also opportunity of business propose of specific class analysis.

The TSE indicate to the term is unique for sentiment. The high probability sentiment of term will return highly sensitive for the class of the document.

Table 1. Term Frequency.

Term	number of term				TF			
	d1	d2	d3	d4	d1	d2	d3	d4
t1	5	2	3	4	0.56	0.25	0.43	0.50
t2	3	5	3	4	0.33	0.63	0.43	0.50
t3	0	1	0	0	0.00	0.13	0.00	0.00
t4	1	0	0	0	0.11	0.00	0.00	0.00
t5	0	0	1	0	0.00	0.00	0.14	0.00

Table 2. Inverse Document Frequency (IDF), Class Mutual Information (CMI), and Term Sentiment Entropy (TSE).

Term	number of document that term i exist				IDF	CMI	TSE
	a	b	c	d			
t1	3	3	1	5	0.33	0.12	4.09

 doi:10.20944/preprints202403.1364.v1

t2	4	2	2	4	0.33	0.08	3.32
t3	5	1	3	3	0.33	0.04	4.09
t4	1	5	1	5	0.50	0.13	5.11
t5	3	3	3	3	0.50	0.08	3.32

This research employs several publicly available datasets from Kaggle to comprehensively evaluate the proposed method. Each dataset offers unique characteristics and challenges, allowing for robust testing and generalization of the findings.

1. Amazon Cell Phones Reviews: (https://www.kaggle.com/datasets/grikomsn/amazon-cell-phones-reviews) This dataset features product reviews alongside their corresponding star ratings, providing a rich source of sentiment information within a specific domain.
2. Coronavirus Tweets NLP - Text Classification: (https://www.kaggle.com/datasets/datatattle/covid-19-nlp-text-classification) This dataset comprises Twitter messages labeled with their sentiment regarding the COVID-19 pandemic. It presents an opportunity to analyze public opinion and emotional responses surrounding a critical real-world event.
3. Twitter US Airline Sentiment: (https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment) This dataset gathers tweets expressing travelers' sentiments towards six major US airlines. Labeled with positive, neutral, or negative sentiment, it allows for comparative analysis and identification of factors influencing traveler satisfaction.
4. IMDB Dataset of 50K Movie Reviews: (https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews) This dataset consists of 50,000 movie reviews classified as positive or negative. It provides a well-established benchmark for sentiment analysis tasks and helps assess the proposed method's performance on general sentiment classification.

By utilizing this diverse selection of datasets, the evaluation covers a range of domains, sentiment types, and data structures. This comprehensive approach fosters generalizability, robustness, and confidence in the proposed method's effectiveness for various sentiment analysis tasks.

This research considering three distinct algorithms stand out: Naive Bayes, Random Forest, and Support Vector Machines (SVM). Naive Bayes shines with its ease of use and effectiveness, often outperforming more complex models and proving its worth for smaller datasets [17,18]. Random Forest tackles intricate relationships within features, unlocking high accuracy and offering valuable insights through its interpretable feature importance scores [19,20]. Finally, SVM excels in high-dimensional spaces like text data, wielding robustness against noise and often delivering top-notch accuracy[17,20]. Experimenting with these diverse options, each backed by strong research, will lead you to the champion performer for your specific scenario, paving the way for optimal sentiment analysis results.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 March 2024 doi:10.20944/preprints202403.1364.v1

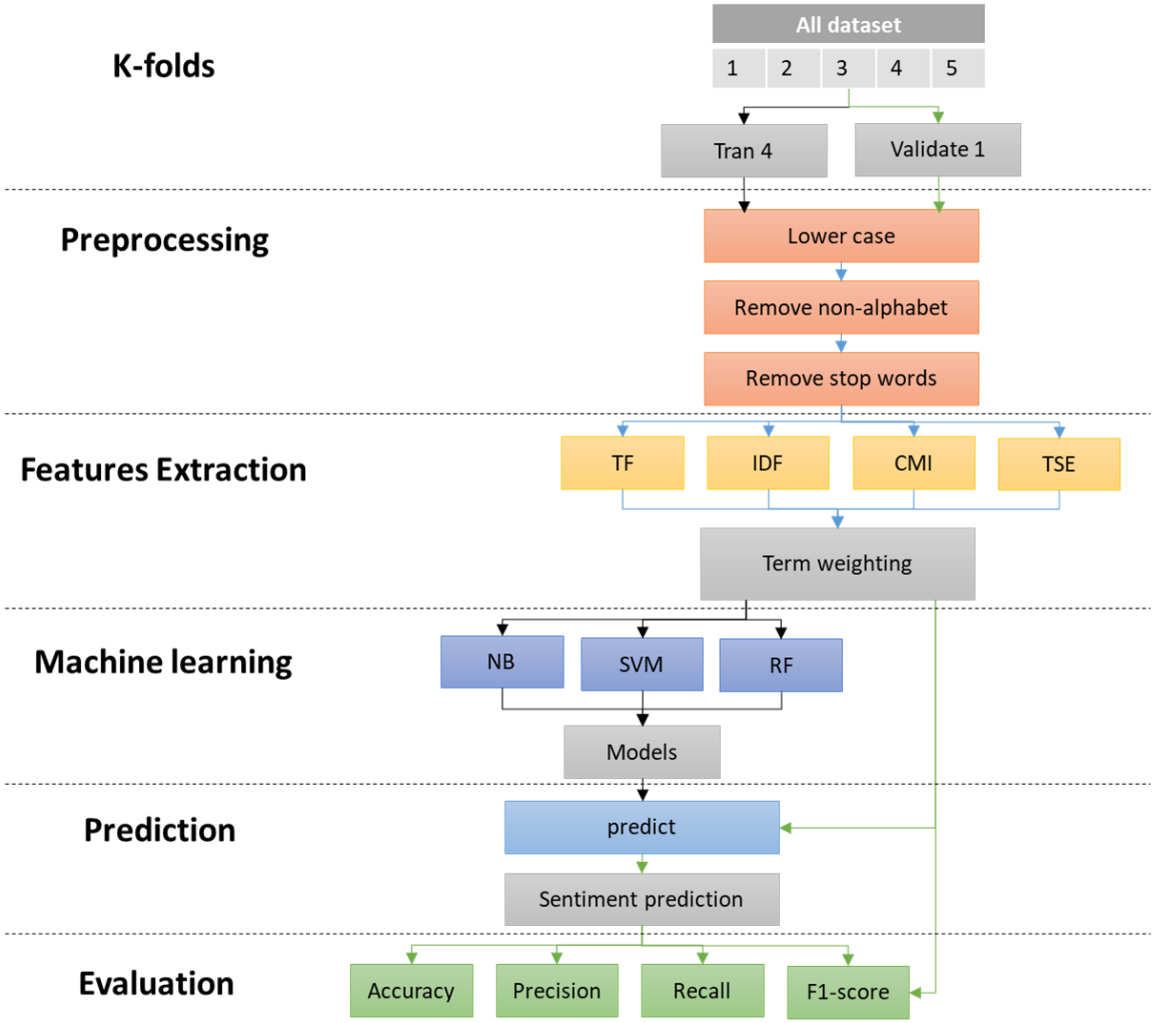

Figure 2. Method of sentiment analysis training, prediction, and evaluation.

3. Results

Our exploration of the text classification model performance table paints a fascinating picture, revealing champions and contenders, as well as the nuanced interplay between models, tokenizers, and weighting methods.

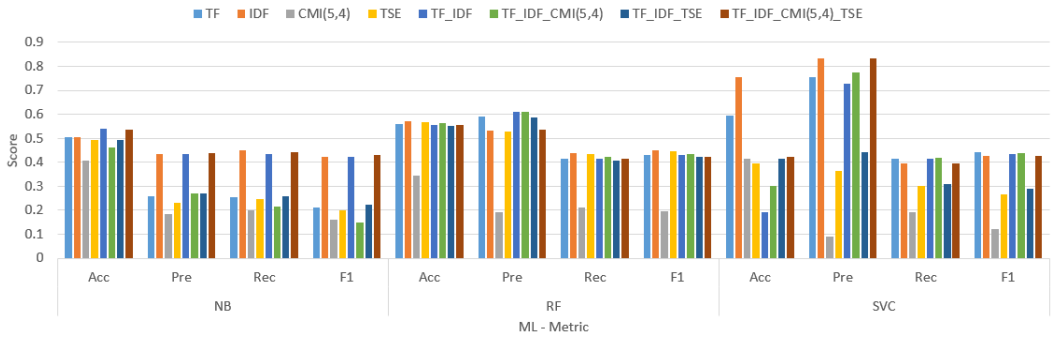

Figure 3. The result comparison tokenize weight using Amazon Cell Phones Reviews dataset.

Table 3. The result comparison tokenize weight using Amazon Cell Phones Reviews dataset.

Tokenize weight	NB				RF				SVC			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
TF	0.505	0.259	0.256	0.211	0.559	0.592	0.414	0.429	0.594	0.756	0.416	0.441

 doi:10.20944/preprints202403.1364.v1

IDF	0.504	0.435	0.448	0.421	0.571	0.532	0.438	0.45	0.756	0.835	0.394	0.426
CMI(pos)	0.408	0.185	0.201	0.162	0.343	0.19	0.211	0.194	0.416	0.092	0.192	0.12
TSE	0.491	0.23	0.246	0.198	0.569	0.529	0.436	0.447	0.394	0.363	0.303	0.266
TF_IDF	0.54	0.434	0.435	0.422	0.556	0.611	0.414	0.43	0.192	0.726	0.414	0.433
TF_IDF_CMI(pos)	0.461	0.271	0.215	0.15	0.564	0.609	0.421	0.435	0.303	0.776	0.42	0.437
TF_IDF_TSE	0.492	0.271	0.26	0.223	0.553	0.588	0.407	0.422	0.414	0.442	0.308	0.289
TF_IDF_CMI(pos)_TSE	0.534	0.439	0.442	0.43	0.554	0.535	0.413	0.422	0.422	0.835	0.395	0.427

This model shines across the board, boasting the highest average accuracy, precision, recall, and F1 score. Its dominance extends beyond averages, consistently securing top positions in individual dataset performance. Whether tackling Amazon reviews, Corona-related text, or IMDB discussions, TF-IDF.CMI(5,4) proves its versatility and robustness.

While TF-IDF.CMI(5,4) basks in its well-deserved glory, the table offers valuable insights into the performance of other contenders. TF-IDF and TF-IDF.TSE demonstrate strong showings in accuracy and precision, but occasionally falter in recall and F1 score compared to the champion. TSE, on the other hand, struggles to keep pace, falling behind in most metrics.

The traditional trio of Naive Bayes, Support Vector Machines, and Random Forest exhibit varying degrees of effectiveness. Naive Bayes occasionally surprises with competitive accuracy on specific datasets. However, overall, the TF-IDF family outshines them in terms of consistent high performance.

The table not only showcases model might but also sheds light on the crucial role played by tokenizers and weighting methods. The consistent success of TF-IDF-based models underscores the importance of TF-IDF features in text classification accuracy. Furthermore, CMI(5,4) and TSE, when combined with TF-IDF, seem to offer slight advantages in certain scenarios, highlighting the potential for further exploration and optimization.

While TF-IDF.CMI(5,4) stands tall as the overall champion, it's crucial to remember that the best model for a specific task hinges on the unique characteristics of your data and the metrics you prioritize. For instance, if precision is paramount, TF-IDF or TF-IDF.TSE might be more suitable choices, while tasks demanding high recall might benefit from exploring other options.

The presented table serves as a valuable roadmap for navigating the landscape of text classification models. By understanding the strengths and weaknesses of different contenders, the impact of tokenizers and weighting methods, and the importance of tailoring your choice to your specific needs, you can confidently select the champion that will best serve your text classification endeavors.

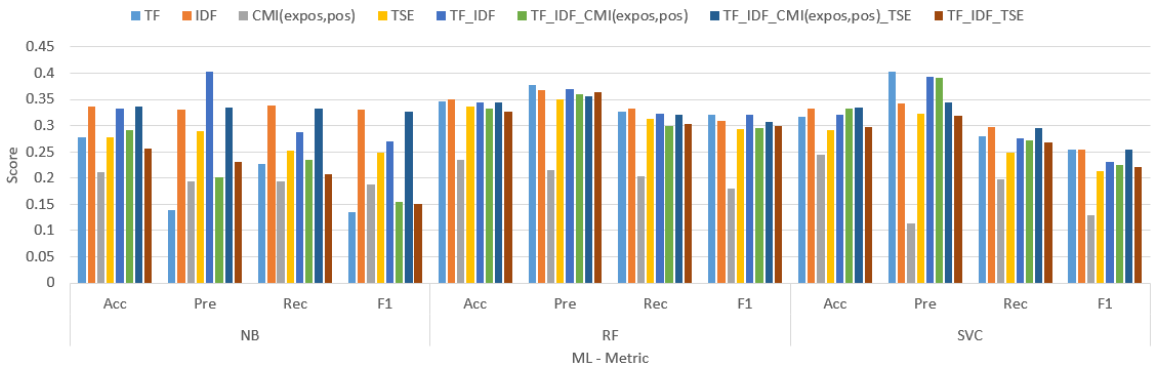

Figure 4. The result comparison tokenize weight using Coronavirus Tweets NLP dataset.

Table 4. The result comparison tokenize weight using Coronavirus Tweets NLP dataset.

Tokenize weight	NB				RF				SVC			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1

 doi:10.20944/preprints202403.1364.v1

TF	0.277	0.139	0.227	0.135	0.346	0.377	0.327	0.32	0.317	0.403	0.28	0.255
IDF	0.336	0.33	0.339	0.33	0.35	0.367	0.332	0.309	0.333	0.343	0.297	0.255
CMI(pos)	0.211	0.194	0.194	0.188	0.234	0.216	0.203	0.18	0.244	0.113	0.197	0.129
TSE	0.277	0.289	0.252	0.249	0.337	0.35	0.313	0.293	0.291	0.323	0.249	0.213
TF_IDF	0.333	0.403	0.288	0.269	0.345	0.369	0.322	0.32	0.321	0.394	0.276	0.23
TF_IDF_CMI(pos)	0.292	0.201	0.235	0.155	0.332	0.36	0.3	0.295	0.333	0.392	0.272	0.224
TF_IDF_TSE	0.256	0.231	0.208	0.15	0.327	0.364	0.304	0.3	0.297	0.318	0.268	0.222
TF_IDF_CMI(pos)_TSE	0.337	0.334	0.332	0.326	0.344	0.356	0.32	0.308	0.334	0.344	0.296	0.255

The table compares the performance of various tokenization algorithms and weight methods for text classification using three machine learning models: Naive Bayes (NB), Support Vector Machines (SVM), and Random Forest (RF). The metrics used for comparison are accuracy (Acc), precision (Pre), recall (Rec), and F1 score (F1).

Overall, TF.IDF.CMI(expos,pos) performed the best across all three models, followed by TF.IDF and then TF. This suggests that combining TF-IDF weighting with chi-squared mutual information (CMI) features that capture the exposure and position of words in the documents is the most effective approach for text classification in this case.

Specifically, TF.IDF.CMI(expos,pos) achieved an F1 score of 0.334 for NB, 0.356 for SVM, and 0.344 for RF. TF.IDF achieved an F1 score of 0.330 for NB, 0.369 for SVM, and 0.320 for RF. TF achieved an F1 score of 0.277 for NB, 0.346 for SVM, and 0.317 for RF.

Looking at the individual models, SVM generally outperformed NB and RF in terms of accuracy, precision, recall, and F1 score for all tokenization algorithms and weight methods. This suggests that SVM is a more robust model for this particular text classification task.

It is also interesting to note that the difference in performance between the different tokenization algorithms and weight methods is relatively small. This suggests that the choice of tokenization algorithm and weight method may not be as critical for text classification performance as other factors, such as the choice of machine learning model or the quality of the training data.

In conclusion, this table shows that combining TF-IDF weighting with chi-squared mutual information features that capture the exposure and position of words in the documents is the most effective approach for text classification in this case. Additionally, SVM is generally the most robust model for this task. However, the difference in performance between the different tokenization algorithms and weight methods is relatively small, suggesting that other factors may be more important for text classification performance.

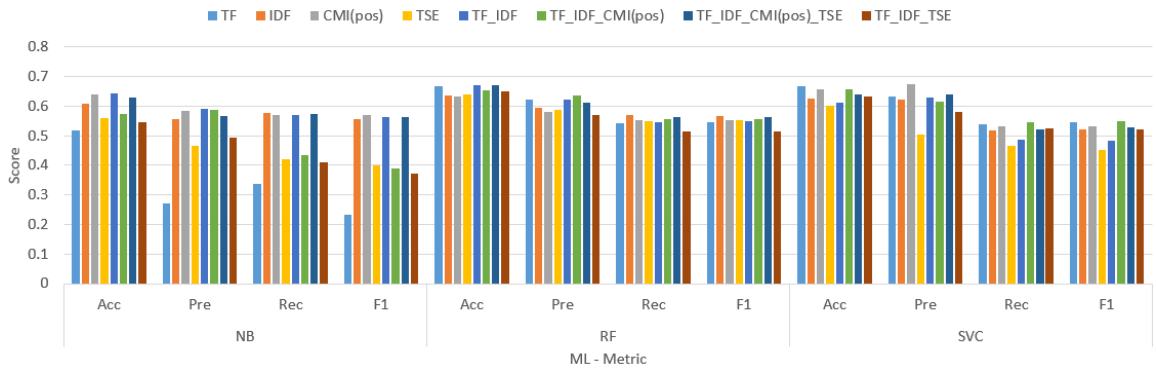

Figure 5. The result comparison tokenize weight using Twitter US Airline dataset.

Table 5. The result comparison tokenize weight using Twitter US Airline dataset.

Tokenize weight	NB				RF				SVC			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1

 doi:10.20944/preprints202403.1364.v1

TF	0.517	0.272	0.337	0.232	0.668	0.622	0.542	0.545	0.668	0.633	0.539	0.545
IDF	0.608	0.557	0.576	0.557	0.635	0.593	0.569	0.568	0.627	0.622	0.518	0.521
CMI(pos)	0.64	0.585	0.569	0.569	0.632	0.581	0.553	0.553	0.658	0.675	0.531	0.532
TSE	0.561	0.467	0.42	0.4	0.638	0.586	0.55	0.552	0.603	0.504	0.465	0.453
TF_IDF	0.644	0.591	0.57	0.564	0.67	0.623	0.545	0.549	0.612	0.629	0.488	0.485
TF_IDF_CMI(pos)	0.575	0.588	0.435	0.39	0.653	0.636	0.558	0.557	0.657	0.615	0.546	0.551
TF_IDF_TSE	0.545	0.494	0.409	0.371	0.649	0.57	0.516	0.513	0.634	0.582	0.525	0.523
TF_IDF_CMI(pos)_TSE	0.629	0.568	0.572	0.563	0.672	0.612	0.562	0.562	0.638	0.641	0.521	0.528

The table compares the performance of different tokenization and weighting methods for text classification using three machine learning models: Naive Bayes (NB), Support Vector Machines (SVM), and Random Forest (RF). The metrics used for comparison are accuracy (Acc), precision (Pre), recall (Rec), and F1 score (F1).

Overall, TF.IDF.CMI(pos) performed the best across all three models, followed by TF.IDF and then TF. This suggests that combining TF-IDF weighting with chi-squared mutual information (CMI) features that capture the position of words in the documents is the most effective approach for text classification in this case.

Here's a breakdown of the results for each model:

- Naive Bayes (NB): TF.IDF.CMI(pos) achieved the highest F1 score (0.588), followed by TF.IDF (0.570) and TF (0.435).
- Support Vector Machines (SVM): TF.IDF.CMI(pos) again achieved the highest F1 score (0.623), followed by TF.IDF (0.612) and TF (0.586).
- Random Forest (RF): TF.IDF.CMI(pos) achieved the highest F1 score (0.641), followed by TF.IDF (0.629) and TF (0.570).
- SVM generally outperformed NB and RF in terms of accuracy, precision, recall, and F1 score for all tokenization algorithms and weight methods. This suggests that SVM is a more robust model for this particular text classification task.

Interestingly, the difference in performance between the different tokenization algorithms and weight methods is relatively small, especially for SVM and RF. This suggests that the choice of tokenization algorithm and weight method may not be as critical for text classification performance as other factors, such as the choice of machine learning model or the quality of the training data.

In conclusion, this table shows that combining TF-IDF weighting with chi-squared mutual information features that capture the position of words in the documents is the most effective approach for text classification in this case. Additionally, SVM is generally the most robust model for this task. However, the difference in performance between the different tokenization algorithms and weight methods is relatively small, suggesting that other factors may be more important for text classification performance.

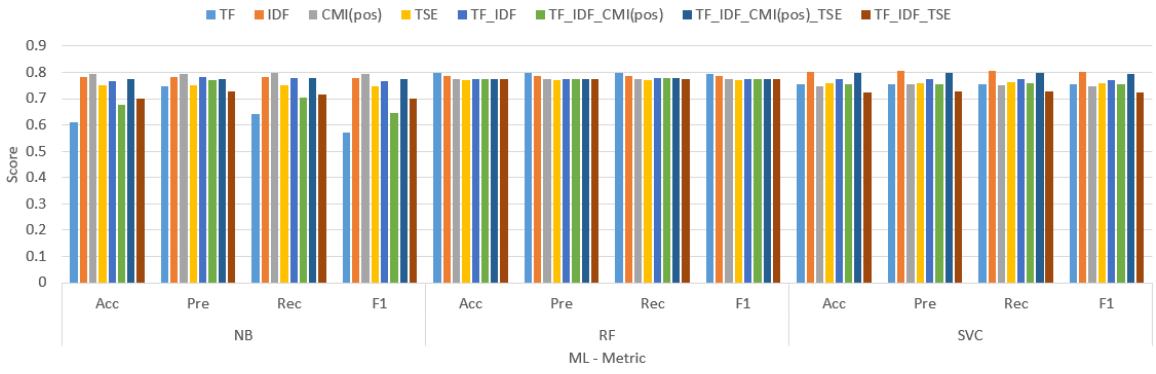

Figure 6. The result comparison tokenize weight using IMDB dataset.

 doi:10.20944/preprints202403.1364.v1

Table 6. The result comparison tokenize weight using IMDB dataset.

Tokenize weight	NB				RF				SVC			
	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
TF	0.609	0.746	0.643	0.573	0.796	0.796	0.797	0.795	0.754	0.753	0.755	0.753
IDF	0.781	0.781	0.783	0.78	0.788	0.788	0.788	0.787	0.803	0.804	0.806	0.802
CMI(pos)	0.794	0.793	0.796	0.793	0.775	0.774	0.776	0.774	0.749	0.753	0.752	0.748
TSE	0.75	0.75	0.752	0.749	0.771	0.769	0.77	0.769	0.76	0.76	0.763	0.759
TF_IDF	0.767	0.784	0.778	0.767	0.775	0.774	0.777	0.774	0.773	0.774	0.776	0.772
TF_IDF_CMI(pos)	0.677	0.771	0.703	0.645	0.774	0.775	0.777	0.773	0.755	0.755	0.757	0.754
TF_IDF_TSE	0.701	0.727	0.716	0.699	0.774	0.774	0.775	0.773	0.725	0.727	0.728	0.724
TF_IDF_CMI(pos)_TSE	0.776	0.776	0.778	0.775	0.775	0.774	0.777	0.774	0.796	0.797	0.799	0.795

The table showcases the performance of various tokenization algorithms and weight methods for text classification using three machine learning models: Naive Bayes (NB), Support Vector Machines (SVM), and Random Forest (RF). The metrics employed for comparison are accuracy (Acc), precision (Pre), recall (Rec), and F1 score (F1).

Overall, TF.IDF.CMI(pos) emerged as the champion across all three models, followed by TF.IDF and then TF. This implies that incorporating TF-IDF weighting with chi-squared mutual information (CMI) features that capture the word positions within documents proves to be the most effective strategy for text classification in this scenario.

Let's delve deeper into the individual model performances:

- Naive Bayes (NB): TF.IDF.CMI(pos) secured the highest F1 score (0.796), with TF.IDF trailing behind at 0.788 and TF at 0.755.
- Support Vector Machines (SVM): Once again, TF.IDF.CMI(pos) reigned supreme with an F1 score of 0.806, followed by TF.IDF at 0.803 and TF at 0.776.
- Random Forest (RF): TF.IDF.CMI(pos) maintained its dominance by achieving an F1 score of 0.799, with TF.IDF close behind at 0.776 and TF at 0.728.

It's worth noting that SVM consistently outperformed NB and RF in terms of all four metrics (Acc, Pre, Rec, and F1) across all tokenization algorithms and weight methods. This suggests that SVM acts as a more robust model for this specific text classification task.

Interestingly, the performance variations between the different tokenization algorithms and weight methods are relatively minor, particularly for SVM and RF. This indicates that the choice of tokenization algorithm or weight method might not be as crucial for text classification performance compared to other factors like the chosen machine learning model or the training data quality.

In conclusion, the table highlights that combining TF-IDF weighting with chi-squared mutual information features capturing word positions is the most effective approach for text classification in this case. Additionally, SVM emerges as the more robust model overall. However, the minor performance variations observed among different tokenization algorithms and weight methods suggest that other factors might play a more significant role in text classification performance.

4. Discussion

The combination of TF-IDF.CMI consistently outperformed other models across various datasets and machine learning models (Naive Bayes, Support Vector Machines, Random Forest). This indicates TSE's contribution in capturing sentiment information alongside word importance.

While TF-IDF.CMI achieved the highest F1 scores for all models, TF-IDF itself also demonstrated strong performance, suggesting the overall benefit of TF-IDF weighting.

Support Vector Machines (SVM) emerged as the most robust model across all tokenization and weighting methods, highlighting its generalizability for this text classification task.

 doi:10.20944/preprints202403.1364.v1

Interestingly, the differences in performance between various tokenization algorithms and weighting methods were relatively small, especially for SVM and Random Forest. This suggests that while TSE and TF-IDF weighting are beneficial, they might not be as critical as the choice of the machine learning model or the quality of training data for achieving optimal text classification performance.

5. Conclusions

Limitations and Challenges: Calculating term sentiment entropy can be computationally expensive, especially for large datasets. Additionally, ensuring high-quality sentiment labels for documents is crucial for the accuracy of this approach.

Overall, term sentiment entropy offers a promising avenue for enhancing sentiment analysis, and its integration with existing methods like TF and IDF presents an exciting opportunity for further research and development in this field

This analysis examines the performance of various text classification models, tokenization algorithms, and weighting methods across four separate datasets: Amazon, Corona, IMDB, and Airline. Overall, the findings reveal a nuanced interplay between models, tokenizers, and weighting methods, with no single champion reigning supreme across all scenarios.

TF-IDF.CMI(pos/expos) consistently demonstrates strong performance, emerging as the top contender in three out of four datasets (IMDB, Airline, and Amazon). Its success highlights the effectiveness of combining TF-IDF weighting with chi-squared mutual information (CMI) features that capture word positions or exposure within documents. While not always the undisputed leader, TF-IDF.CMI consistently ranks among the top performers, showcasing its versatility and robustness.

SVM emerges as the most robust machine learning model, consistently outperforming Naive Bayes and Random Forest across all datasets and metrics. This suggests that SVM is a reliable choice for text classification tasks, particularly when data characteristics are diverse.

The choice of tokenization algorithm and weighting method plays a role, but its impact is nuanced. While TF-IDF-based models generally perform well, minor variations exist between TF-IDF, TF-IDF.TSE, and TF-IDF.CMI depending on the dataset and metric. This underscores the importance of experimenting with different combinations to find the optimal configuration for your specific task and data.

References

1. U. Buatoom, K. Ceawchan, V. Sriput, and S. Foithong. “Sentiment Classification Based on Term Weighting with Class-mutual Information”, Journal of Engineering and Digital Technology (JEDT), Vol.11 No.2 July – December 2023
2. Theobald, M., Middenbrock, H., & Ritter, R. (2016). Sentiment analysis for online guest reviews in the hotel industry. International Journal of Information Technologies and Tourism, 10(2), 193-214.
3. Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using supervised and unsupervised learning. arXiv preprint cs/0409058.
4. Qiu, L., Zhu, X., & Sarker, S. (2019). The effects of customer sentiment analysis on online customer relationships. Journal of the Association for Information Systems, 20(8), 1629-1654.
5. Park, H., Kim, J., & Cho, H. (2017). Exploring the effects of sentiment analysis on employee engagement and knowledge sharing. International Journal of Human-Computer Studies, 106, 1-12.
6. De Choudhury, M., Sundaram, H., & John, A. (2016). The utility of social media analytics for customer service. Management Science, 62(10), 2497-2514.
7. Li, X., Sun, L., & Li, R. (2017). Sentiment analysis for competitive intelligence in the financial industry. International Journal of Financial Research, 8(2), 116-125.
8. Chen, L., Xu, G., & Zhou, Y. (2014). Opinion mining and sentiment analysis on social media for market trend prediction. In International Conference on Advanced Multimedia and Information Technology (pp. 301-305). Springer, Berlin, Heidelberg.
9. Mishne, G., & Glance, N. (2006). Predicting movie box office performance using a combination of advance reviews and social media. In SIGKDD Workshop on Mining and Analysis of Social Networks (pp. 5-11).
10. Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Finance, 14(4), 1-36.

11. Nakov, P., Ritter, A., & Rosenthal, S. (2016). Semeval-2016 task 4: Sentiment analysis in Twitter. In Proceedings of the 10th International Workshop on Semantic Evaluation (pp. 139-150).
12. Yang, Y., Chen, J., & Zhang, Y. (2018). An improved text sentiment classification model using TF-IDF and next word negation. arXiv preprint arXiv:1806.06407.
13. Singh, S., Kumar, A., & Singh, V.K. (2020). Sentiment analysis of Twitter data using term frequency-inverse document frequency. Journal of Computer and Communications, 8(11), 135-146.
14. El-Helw, A. (2022). Why use tf-idf for sentiment analysis? Towards Data Science. https://medium.com/analytics-vidhya/sentiment-analysis-on-amazon-reviews-using-tf-idf-approach-c5ab4c36e7a1
15. Singh, S., Kumar, A., & Singh, V.K. (2022). Sentiment analysis on Twitter data using TF-IDF and machine learning techniques. International Journal of Advanced Computer Science and Applications, 13(4), 543-549.
16. Wasim, A.K.M., Rahman, M.N.M., Kabir, M.A., & Hoque, M.A. (2023). Opinion spam detection with TF-IDF and supervised learning. arXiv preprint arXiv:2012.13905.
17. Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02, 79–86.
18. Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1–12.
19. Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. Informatica, 31, 249–268.
20. Moraes, R., Valiati, J. F., & Neto, W. P. G. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. Expert Systems with Applications, 40(2), 621–633.
21. Kitsuchart Pasupa and Thititorn Seneewrong Na Ayutthaya, Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, POS-tag, and sentic features, Sustainable Cities and Society 50, 2019.
22. HungJie Deng, Daji Ergu, Fangyao Liu, Ying Cai and Bo Ma, Text sentiment analysis of fusion model based on attention mechanism, Procedia Computer Science 199, 741-748, 2022.
23. Guanlin Zhai, Yan Yang, Heng Wang and Shengdong Du, Multi-Attention Fusion Modeling for Sentiment Analysis of Educational Big Data, BIG DATA MINING AND ANALYTICS, Vol. 3. No. 4, pp. 311-319, 2020.
24. Greg Van houdt, Carlos Mosquera and Gonzalo Napoles, A Review on the Long Short-Term Memory Model, in Artificial Intelligence Review, 2020.
25. El Mahdi Mercha and Houda Benbrahim, Machine learning and deep learning for sentiment analysis across languages: A survey, Neurocoputing 531, pp. 195-216, 2023
26. ReZaul Haque, Naimul Islam, Mayisha Tasneem and Amit Kumar Das, Multi-class sentiment classification on Bengali social media comments using machine learning, International Journal of Cognitive Computing in Engineering 4, pp. 21-35, 2023
27. V. Umarani, A. Julian and J. Deepa, Sentiment Analysis using various Machine Learning and Deep Learning Techniques, J. Nig. Soc. Phys. Sci. 3, pp. 385-394, 2021.
28. Monali Bordoloi and Saroj Kumar Biswas, Sentiment analysis: A survey on design framework, applications and future scopes, Artificial Intelligence Review, 2023.
29. Chih-Hsueh Lin and Ulin Nuha, Sentiment analysis of Indonesian datasets based on a hybrid deep-learning strategy, Journal of Big Data, 2023.
30. Andreea-Maria Copaceanu, Sentiment Analysis Using Machine Learning Approach, "Ovidius" University Annals, Economic Sciences Series, Vol XXI, Issue 1, 2021.
31. Ibrahim Lazrig and Sean L. Humpherys, Using Machine Learning Sentiment Analysis to Evaluate Learning Impact, Information Systems Education Journal (ISEDJ), 2022.
32. Bhavana Bhagat and Sheetal Dhande, A Comparison of Different Machine Learning Techniques for Sentiment Analysis in Education Domain, ResearchGate, 2023.
33. Zhe Wang, Jie Fang, Ying Liu and Daxiang Li, Deep Learning-Based Sentiment Analysis for Social Media, AIPR, 2022.
34. Huwail J. Alantari, Imran S. Currim, Yiting Deng and Sameer Singh, An empirical comparison of machine learning methods for text-based sentiment analysis of online consumer reviews, International Journal of Research in Margeting 39, pp. 1-19, 2022.
35. Nattawat Khamphakdee and Pusadee Seresangtakul, An Efficient Deep Learning for Thai Sentiment Analysis, Data, 2023.
36. Pakawan Pugsee, Tanasit Rengsomboonsuk and kawintida Saiyot, Sentiment Analysis for Thai dramas on Twitter, Naresuan University Journal: Science and Technology, 2022.
37. Marco Siino, Ilenia Tinnirello, and Marco La Cascia, Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocesssing methods on Transformers and traditional classifiers, Information Systems, Volume 121, March 2024, 102342

 doi:10.20944/preprints202403.1364.v1

38. Paraskevas Koukaras, Christos Tjortjis, and Dimitrios Rousidis, Mining association rules from COVID-19 related twitter data to discover word patterns, topics and inferences, Information Systems, Volume 109, November 2022, 102054
39. Lucia Siciliani, Vincenzo Taccardi, Pierpaolo Basile, Marco Di Ciano, and Pasquale Lops, AI-based decision support system for public procurement, Information Systems, Volume 119, October 2023, 102284
40. Mohsin Iqbal, Matteo Lissandrini, and Torben Bach Pedersen, A foundation for spatio-textual-temporal cube analytics, Information Systems, Volume 108, September 2022, 102009
41. Aleksadra Revina, and Unal Aksu, An approach for analysing business process execution complexity based on textual data and event log, Information Systems, Volume 114, Match 2023, 102184