Term-Recency for TF-IDF, BM25 and USE Term Weighting

Divyanshu Marwah
School of Computer Science
and Statistics
Trinity College Dublin
Dublin, Ireland
marwahd@tcd.ie

Joeran Beel
School of Computer Science
and Statistics
Trinity College Dublin
Dublin, Ireland
joeran.beel@scss.tcd.ie

Abstract

Effectiveness of a recommendation in an Information Retrieval (IR) system is determined by relevancy scores of retrieved results. Term weighting is responsible for computing the relevance scores and consequently differentiating between the terms in a document. However, current term weighting formula like TF-IDF weigh terms only based on term frequency and inverse document frequency irrespective of other important factors. This results in uncertainty in cases when both TF and IDF values are same for more than one document, hence resulting in same term weight values. In this paper, we propose a modification of TF-IDF and other term-weighting schemes that weights terms additionally based on the recency of a term, i.e. the metric based on the year the term occurred for the first time and the document frequency. We modified the term weighting schemes TF-IDF, BM25 and Universal Sentence Encoder (USE) to additionally consider the recency of a term and evaluated them on three datasets. Our modified TF-IDF outperformed the standard TF-IDF on all three datasets; the modified USE outperformed the standard USE on two of the three datasets; the modified BM25 did not outperform the standard BM25 term-weighting scheme.

1 Introduction

Term Weighting is one of the most crucial tasks in information retrieval and recommender systems. It is method of quantifying terms in a document to determine the importance of the words in the document and the corpus (El-Khair, 2009). Apart from recommendation engine and information retrieval, term weighting is effective in many scenarios such as text mining, text classification, duplicate image detection (Chum et al., 2008), document clustering, and even in medical science research. In text categorization and data mining, efficient. term weighting brings a considerable boost in effectiveness (Domeniconi et al., 2015). Several term weighting approaches are used in different applications basically derived from the frequency and distribution of words in documents (Domeniconi et al., 2015).

TF-IDF is one of the classic term weighting approaches, that is most frequently used and was found to be used, for instance, by 83% of text-based research paper recommender systems (Beel et al., 2017). TF-IDF as the name suggests, is made up of two parts, term frequency (TF) and inverse document frequency (IDF). TF gives the number of times a term occurs in a document. The basis is that the more frequently a term occurs, the more it is important for the context of the document (Beel et al., 2017). IDF is computed as the inverse frequency of documents containing the searched term. The idea behind this is that a rare term should be given higher importance as compared to frequently occurring terms such as articles, pronouns, etc. There have been numerous researches on TF-IDF, and many extensions and alternatives are suggested. Some other term weighting models used are BM25, LM Dirichlet, Divergence from independence, etc. Text and sentence embedding models such as Universal Sentence Encoder (USE) (Cer et al., 2018), Google's BERT, InferSent, etc are also used in text classification tasks. These different approaches depend on the type, size of corpus, types of queries, and they use different term metrics to determine the effectiveness of term in a document and corpus.

In case of information retrieval task, there are certain limitations in standard term weighting approaches. Analyzing the simple approach of TF-IDF, that weights term based on the frequency distribution in the corpus. The real issue in this method is the assumption that frequency distribution remains constant with time, without contemplating the diverse contexts for different terms. In short periods, this holds, however, over longer time this assumption fails. For example, consider two terms, ”COVID19” and ”neural networks”, that have different origin years. Now, there are probably fewer documents containing the term ”COVID19” than documents containing the term ”neural networks”, simply because ”COVID19” is a relatively new term, while ”neural networks” is a term being used since decades. However, they would be weighted similarly without considering the difference in the origins. The issue that terms have temporal distributions of frequency, not just space distribution is unaccounted when using the standard term weighting methodologies.

Considering this uncertainty in term weighting, we suggest a time-normalized term weighting approach, which reflects the age of a term. As the vocabulary changes over time, our intuition is to identify a term’s age based on its first usage and current year and distinguish between the documents based on the age of the terms used. Hence, we propose to weigh terms not only on their frequency distributions but also temporal distributions. Furthermore, we demonstrate the significance of adding a time-based feature by comparing our method with state-of-the-art baseline models, that is, TF-IDF, BM25 and USE embedding. Experimental results show substantial improvements over the baseline models for similar recommendations.

2 Related Work

TF-IDF is a relatively old approach and there have been many studies comparing the results of TF-IDF with other states of the art term weighting schemes. Also, different researches have suggested novel variants and enhancing algorithms solving various issues. For instance, (Beel et al., 2017) points out the lack of personalization in classic TF-IDF. The authors have highlighted the issue of access to the document corpus for calculating IDF and another issue of ignoring the information from the user’s document collection for recommendations and user modelling. Thus, a novel term weighting is suggested, that does not require the document corpus and uses the user’s document collection for user modelling.

In another paper, (Domeniconi et al., 2015) points out the problem of using IDF in text classification. The basic idea behind IDF is that a term occurring frequently has negligible distinguishing power, however, in the case of text classification, this might not be true, because, highly frequent terms in different documents of the same category can be helpful in text classification. Hence, the authors suggested a supervised learning approach to calculate IDF excluding the category under consideration.

(Park et al., 2005) suggests a novel approach to term weighting based on the term positions along with the TF and IDF terms. The authors studied the term patterns that occur in the documents using the wavelet transform method. The paper also suggests that the documents are ranked more relevant if the query terms are close to each other.

Utilizing temporal feature has also proved to be an efficient way for recommenders and time normalized recommendations are certainly receiving growing application in recent times (Campos et al., 2014). One of the researches (Kacem et al., 2014), suggests usage of time-normalized term weighting for user modelling. The authors have used the time of social/web search of terms to form the short and long-term contexts and further creating a user profile based on the same. The comparative study of this algorithm with the standard TF-IDF suggests a significant improvement in results centered on the time normalized user models. Considering this research of temporal context’s effect on term weights, we propose a Time Normalized TF-IDF algorithm for information retrieval and recommender system, discussed and implemented in this paper.

3 Time Normalized Term Weighting

In a classic term weighting approach, the terms are weighted irrespective of the different contexts or usage or recency. In this paper, we try to emphasize on the importance of term recency in relevant results retrieval. The premise for this algorithm is that, if a term is devised newly then there are probably a lesser number of documents containing the term compared to the term which is being used for a longer duration of time. In this algorithm, we introduce a time factor along with the regular TF-IDF values. This time-based factor is formulated from the origin year of the word and the document frequency of the term giving the metric as documents per year. For a given term w in document $d \epsilon D$, where D is the document corpus D with size N, term-age is calculated as:

$$t_{w,D} = \log(df_{w,D}/(y_{diff}+1))^{1}, \qquad (1)$$

[1] This is an updated formula with an added 1 in the denominator, for our experiments, we used the older version of formula

where y_{diff} is calculated as :

$$y_{diff} = y_{current} - y_{origin} \tag{2}$$

where y_{origin} is the year of first usage of the word and $Y_{current}$ is the present year. This current year remains constant in the calculations, giving us the sort of age for the word. We take the logarithm of the terms to normalize the value, since this can go to a large number based on the size of the corpus. Also, we take up the absolute value of log, so that we don't have negative weight values. The y_{origin} can be traced from multiple places depending on the problem statement. For example, if a research paper recommender is being developed, the origin year can be retrieved as the year of first occurrence of the term in the recommendation corpus. Or in case of web search, time of first search of the term can be used. Likewise, for some instances the terms can be traced to their etymology and the year of first occurrence can be fetched. Now the updated formula for term weight calculation for tTF-IDF is given as:

$$wt_{w,d} = t_{w,D} * tf_{w,d} * log(\frac{N}{df_{w,d}}) \tag{3}$$

where $t_(w, D)$ is the time-factor calculated value in equation (1) $tf_(w, d)$ is the number of times term w occurs in a document d, and $df_(w, D)$ is the number of documents in which w appears in D.

Likewise, in case of time normalized BM25 (tBM25) model, the term age, $t_(w, D)$ is multiplied to the classic BM25 formula for the time normalized model. For USE embedding approach, cosine similarity is used to calculate the term weights. In the time normalized model, we multiply the term age factor, $t_(w, D)$ with the cosine similarity function to get the updated time normalized USE (tUSE) model.

Now, assume that a term is new and occurs in reasonable number of documents, then the value of $t_(w, D)$ will be large and hence the term weight will be large. Similarly, if the term is being used for many years and is occurring in many documents, it will relatively reduce the value of the time-factor, thus giving it low importance.

A caution which needs to be taken while implementing this algorithm is to check for more commonly occurring non relevant terms which are normalized by using IDF should not get boosted. This can be taken care of while calculating the value of y_{origin}, and such terms can be ignored so they don't boost up the term weights based on non-relevant terms.

4 Implementation

4.1 Data

4.1.1 TREC Washington Post Corpus (Post, 2018)

This collection contains 608,180 news articles and blog posts, along with 50 queries from TREC – 2018 news background linking task (Soboroff et al., 2018), and expected set of results. For the purpose of testing our hypothesis, we use a sample of 20909 documents with approximately 2400 relevant documents. However, this has been done only for time-based index due to scalability and resource constraints. And the term age is still calculated considering the entire corpus and does not affect the algorithmic logic. The relevant fields in this dataset are id, URL, title, author, and article text.

4.1.2 Web Answer Passage(WebAP) Dataset (Keikha et al., 2014, 2015)

This collection contains 8027 articles from the web, which are answers to 82 TREC queries. The dataset contains the following fields: unique document id, target question id, and passage. The results contain 50 relevant documents, given as question_id, document number and relevance as ranked from 1 to 50.

4.1.3 CiteULike Dataset (Wang et al., 2013)

This dataset is collected from CiteULike and Google Scholar and contains 17013 documents with the following fields: document id, title of the research paper, and abstract. We are given another file in this dataset, that contains the referenced articles for every document. We have randomly selected 116 test topics having exactly 10 citations to be used as our ground truth.

4.2 Architecture/Methodology

We implemented text-based recommendation systems using the data mentioned in the last section. This is implemented using TF-IDF, BM25 and USE embedding models. Further, we devised an algorithm to calculate $t_(w, D)$ as described earlier. And scoring is done using customized plugins. Finally, we compare the results of different algorithms using the evaluation metrics described later.

The first index is created with the same mapping structure as given in the input dataset files. For the second index, we add a time normalized term weight parameter as a payload to the terms while indexing the documents. A third index is created using the time normalized index for USE embedding model. In this index, we calculate the term vectors using the pre-trained TensorFlow model (Cer et al., 2018) and store it in a 512-length vector field. This methodology remains the same for all the datasets. Following steps are used for calculations of term age:

- Consider the article text of the document and fetch the origin year for every word from etymonline.com

- Calculate the difference in number of years from the year of first occurrence, to the current year. We have assumed the base year for our corpus to be 2017(TREC News), 2015(Web AP) and 2019(CiteULike) since that is the year of latest publications. This has been done for uniform term weighting across the corpus.

- Now the term weight is calculated using the formula given in section 3.

An important part to note is, that every term is not given a term weight, this happens if the origin year of the term is not traceable, or the terms are most frequently used such as articles, or prepositions. We have used the following evaluation metrics to evaluate the significance of retrieved results:

- *Precision @10*: We have calculated the precision value for top 10 fetched results on the given set of input queries and take an average of the results for comparison.

- *Recall*: For calculating the recall, we have considered the queries having less than 100 results, in case of TREC news. And then recall is calculated as the number of relevant retrieved document divided by the number of relevant documents present in the index. For other datasets, since the number of relevant results is fixed, so the value of precision and recall remains the same.

- *F1 Score*: Since F1 score uses both the precision and recall values, so for calculating the precision scores, we have used the same results from the recall measure and used a fixed denominator as the number of retrieved results. Formula used for F1 score is given as:

$$F1 = 2 * \frac{Precision * Recal}{Precision + Recall} \tag{4}$$

- *Normalized Discounted Cumulative Gain*: DCG is calculated at specific rank position p, given by

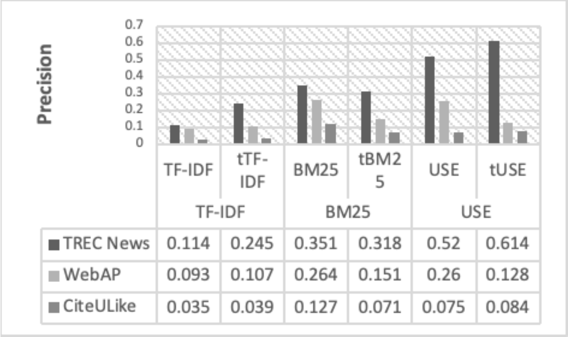
$$DCG_p = \sum_{i=1}^{p} rel_i / \log_2(i+1) \tag{5}$$

where, rel_i is the relevance score of a document at position i. And NDCG is calculated by considering the DCG of ideal order along with the DCG values and is given by

$$nDCG_p = DCG_p / IDCG_p \tag{6}$$

5 Results and Discussion

In 2 out of 3 algorithms, our term-recency modification improved the performance notably. When measured by p@10, tTF-IDF outperformed TF-IDF by an average 47% and tUSE outperformed USE in 2 of the 3 datasets by 14.3% but performed 50% worse in the other dataset (Figure 1). The time normalized BM25 version, however, performed 32% worse than BM25. NDCG@10 leads to similar results (Figure 2). For CiteULike dataset, NDCG cannot be calculated, since there is no ranking specified for the citations.

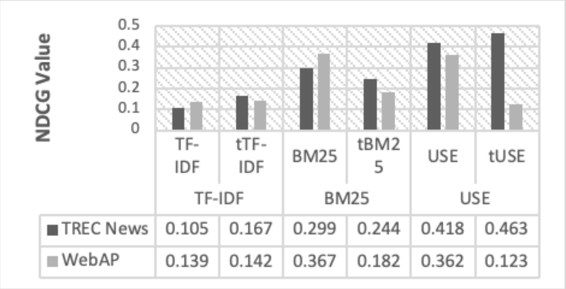


	TF-IDF	tTF-IDF	BM25	tBM25	USE	tUSE
	TF-IDF		BM25		USE	
■ TREC News	0.114	0.245	0.351	0.318	0.52	0.614
■ WebAP	0.093	0.107	0.264	0.151	0.26	0.128
■ CiteULike	0.035	0.039	0.127	0.071	0.075	0.084

Figure 1: P@10 comparison

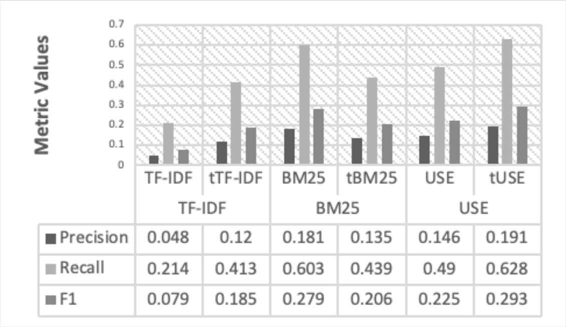
On closer analysis of the BM25 model, we see 27 out of 50 queries gave better or almost similar results in case of time normalized model when compared to the classic approach. These results are also promising and need to be worked upon for better results in the future work.

For calculating the recall and F1 scores, we fetch the top 100 results for the given query sets. For



	TF-IDF		BM25		USE	
	TF-IDF	tTF-IDF	BM25	tBM25	USE	tUSE
■ TREC News	0.105	0.167	0.299	0.244	0.418	0.463
■ WebAP	0.139	0.142	0.367	0.182	0.362	0.123

Figure 2: NDCG@10 comparison



	TF-IDF		BM25		USE	
	TF-IDF	tTF-IDF	BM25	tBM25	USE	tUSE
■ Precision	0.048	0.12	0.181	0.135	0.146	0.191
■ Recall	0.214	0.413	0.603	0.439	0.49	0.628
■ F1	0.079	0.185	0.279	0.206	0.225	0.293

Figure 3: Precision, Recall, F1@100 scores for TREC News

uniformity in the metric calculation, we compute the precision scores as well. The result metrics are shown in Figure 3. We see a 150% improvement in the tTF-IDF model over TF-IDF and a 31% improvement in tUSE model over the USE model. Recall and F1 scores for WebAP and CiteULike would be same as precision scores since the number of relevant results in dataset remains the same, so they are not shown.

Analyzing the tUSE model in WebAP dataset, we see it does not perform well against the USE model. One of the possible reasons for this might be the size of the corpus used, that is, TREC news corpus has approximately 600k documents while Web AP dataset has just 6k documents, which is 100 times less than the former dataset. However, this is an inference based on the results retrieved and has not been verified. There might be other possible reasons, such as the size of documents, size of queries used, number of proper nouns in the queries, etc. Or probably term age might not be a relevant metric for this dataset. These possible reasons still need to be analyzed before affirming out a conclusion on these contrasting results.

6 Conclusion

In this paper, we proposed a novel algorithm for term weighting. The presented approach shows the significance of temporal distribution along with existing space distribution of terms. We suggest the scheming of a term recency parameter based on the origin of the word and the usage in the document corpus. This factor is used along with the standard weighting values (such as TF and IDF) for relevance scoring in the information retrieval system. The algorithm is tested on a news dataset, with queries trying to find links with the documents, Web answer retrieval dataset and research papers citations dataset. We have also extended the algorithm to other text embedding models that are BM25 and USE.

Experiments conducted on the IR system show that term-recency based TF-IDF and tUSE model outperforms the classic TF-IDF and classic USE algorithms with a significant margin when measured in terms of average precision, recall, F1 and NDCG. It has set up a strong premise for our ongoing research on ways to improve recommendation effectiveness. Future works for this can be to find ways to improve the time-based BM25 model and testing the algorithm's performance in other tasks such as text classification, and user modelling. Furthermore, we also plan to test different normalization factors for calculating the term age and then using it in the scoring algorithm.

Acknowledgments

We would like to thank Sneha Srivastava, Rachit Rastogi and Shivam Khanduri for their comments, suggestions and feedback.

References

Joeran Beel, Stefan Langer, and Bela Gipp. 2017. *TF-IDuF: A novel term-weighting scheme for user modeling based on users' personal document collections*. University of Illinois.

Pedro G. Campos, Fernando Díez, and Iván Cantador. 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1-2):67–119.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, and Chris Tar. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Ondrej Chum, James Philbin, and Andrew Zisserman. 2008. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, volume 810, pages 812–815.

Giacomo Domeniconi, Gianluca Moro, Roberto Pasolini, and Claudio Sartori. 2015. A study on term weighting for text categorization: A novel supervised variant of tf. idf. In *DATA*, pages 26–37.

Ibrahim Abu El-Khair. 2009. *Term Weighting*, pages 3037–3040. Springer US, Boston, MA.

Ameni Kacem, Mohand Boughanem, and Rim Faiz. 2014. Time-sensitive user profile for optimizing search personlization. In *International conference on user modeling, adaptation, and personalization*, pages 111–121. Springer.

Mostafa Keikha, Jae Hyun Park, W. Bruce Croft, and Mark Sanderson. 2014. Retrieving passages and finding answers. In *Proceedings of the 2014 Australasian Document Computing Symposium*, pages 81–84.

Mostafa Keikha, Jae Hyun Park, W. Bruce Croft, and Mark Sanderson. 2015. Web answer passages (webap) dataset.

Laurence A. F. Park, Kotagiri Ramamohanarao, and Marimuthu Palaniswami. 2005. A novel document retrieval method using the discrete wavelet transform. *ACM Transactions on Information Systems (TOIS)*, 23(3):267–298.

Washington Post. 2018. Trec washington post corpus.

Ian Soboroff, Shudong Huang, and Donna Harman. 2018. Trec 2018 news track overview. In *The Twenty-Seventh Text RE-trieval Conference (TREC 2018) Proceedings*.

Hao Wang, Binyi Chen, and Wu-Jun Li. 2013. Collaborative topic regression with social regularization for tag recommendation. In *Twenty-Third International Joint Conference on Artificial Intelligence*.