https://doi.org/10.2352/issn.2168-3204.2017.1.0.146

Using a Large Set of Weak Classifiers for Text Analytics

Steven J. Simske and A. Marie Vans; HP Labs; Fort Collins, Colorado/USA

Abstract

*TF*IDF is a common approach used for text mining and information retrieval. We have described a method for using 112 variations on the TF*IDF equation for the classification of 588 CNN news articles belonging to 12 different classes [1]. We found that no single TF*IDF could accurately classify all the documents. In fact, the highest accuracy attainable by any single TF*IDF was 45%. In this article, we take the work further to show how different measurements utilizing the TF*IDF classification results can be used to show that some classes may be logically inconsistent as classes. These methods also may be used to create more cohesive classes.*

Introduction

Determining "aboutness" [1] of a document is the first step in the classification of documents for later retrieval during the search process. "Aboutness" is generally defined by one of two expert readers (1) the author herself, who provides keywords to index the document, or (2) an expert indexer, usually an employee of the publishing organization[2]. What is needed is an automated approach to discover the keywords and key terms using the words contained in the document for classifying documents.

We use a set of TF*IDF (Term Frequency *x* Inverse Document Frequency) measures on 588 tagged CNN news articles [3]. While the use of TF*IDF for extracting topic information from CNN articles is not new [4], our approach is novel in that we define 112 different permutations of the TF*IDF measure and analyze each one individually and in combination for classification purposes. Previously, we have described accuracy and combination experiments [5]. This article focuses on two new measures that use the results of the previous experiments for the purpose of identifying logically inconsistent classes as well as the creation of more cohesive classes. These new measures can also be used to strengthen the previous results which determined the highest performing TF*IDF measures.

In this paper we describe the TF*IDF equations used in the experiment as well as two new metrics: mean_attempts_to_classify and attempt_entropy. We then show the results of using these metrics on the TF*IDF results previously describe and discuss the implications of these results. Finally, we conclude with some ideas on how the work can be taken forward.

Methods and Materials

*TF*IDF*

TF*IDF [6, 7] is commonly used in information retrieval and classification tasks [8, 9, 10]. We have defined a total of 112 TF*IDF equations created by using a combination of 14 inverse document frequency equations for each of 8 term document frequency equations. These were computed for a set of CNN articles, which were assigned to 12 classes. Within each class, articles were assigned to two equally-sized groups: one for training and one for testing. The total number of files used for each class depends on the number of files in the class with the smallest number of files assigned to it. In our case, one class had only 98 files total assigned to it. The largest class contained 988 files. In order to make sure all classes contributed evenly to the classification task, we used selected 49 files chosen randomly, for each training and each test set from each class. Table 1 provides the 8 term frequency (TF) measures while Table 2 provides the 14 inverse document frequency (IDF) measures. To build a measure, we multiply one of the TF measures by one of the IDF equations. For example, the Power-Mean measure would be implemented as shown in Equation 1:

$$\left(w_{i,j}{}^{Power}\right) * \left(N - 1/w_{i,n}\right) \quad (1)$$

An experiment consists of preprocessing each document and creating an input stream for each article. We create a stream of tokens composed of individual words using the sharpNLP [11] C# open source project. The stream is then converted into a bag of words consisting of all non-stop words in each file.

Once the TF*IDF measures are generated for each word in the file, we can create a master list of words for all the files in a given class. We create this master list by summing all the TF*IDF values for each word and dividing this sum by the number of files in which the word is found (normalization). This gives us a single TF*IDF measure for each word found in a class which we can then use for classifying articles from the test set of documents.

During testing, we first determine the TF*IDF measure values for all the words in a document. We then compare them with the normalized values for that word in each of the classes using the dot product of the TF*IDF value for the word in the test file with that of the normalized TF*IDF value for the word in each training class. A high result value indicates that the word may belong to the class. The class that produces the highest dot product values for all the words in the file is then assigned as the class for that document. This procedure is used for all the test files in each class for each of the 112 TF*IDF measures.

We found that no single TF*IDF could accurately classify all the documents [5]. Using different measurements that utilize the TF*IDF classification results described above, we can show that some classes may be impossible to classify. We can also suggest means to better create cohesive classes.

Table 1: TF Equations Used in Experiments (from [5])

	TF Name	TF Numerator
1	**Power**	$(w_{i,j})^{Power}$
2	**Mean**	$w_{i,j}$
3	**NormLog**	$1+\log_2(w_{i,j}) \Big/ \log_2(k)$
4	**Log**	$1+\log_2(w_{i,j})$
5	**NormLogs**	$1+\log_2(w_{i,j}) \Big/ \log_{\frac{2}{LogRatio}}(k)$ $If\ LogRatio \geq MinLogRatio$ $1+\log_2(w_{i,j}) \Big/ \log_2(k)$ $If\ LogRatio < MinLogRatio$
6	**NormMean**	$w_{i,j} \Big/ k$
7	**NormPower**	$(w_{i,j})^{Power} \Big/ k^{Power}$
8	**NormPowers**	$(w_{i,j})^{WordPower} \Big/ k^{DocPower}$

Table 2: IDF Equations Used in Experiments. (from [5])

	IDF Name	IDF Denominator
1	**NormLogsOfSums**	$\log_{\frac{2}{LogRatio}}(\sum_{j=1}^{N-1} k_j) \Big/ (1+\log_2(w_{i,n}))$ *if LogRatio ≥ MinLogRatio* $\log_2(\sum_{j=1}^{N-1} k_j) \Big/ (1+\log_2(w_{i,n}))$ *if LogRatio < MinLogRatio*
2	**NormSumsOfLogs**	$\log_{\frac{2}{LogRatio}}(\sum_{j=1}^{N-1}(k_j)) \Big/ (\sum_{n=1}^{N-1}(1+\log_2(w_{i,n})))$ $If\ LogRatio \geq MinLogRatio$ $\log_2(\sum_{j=1}^{N-1}(k_j)) \Big/ (\sum_{n=1}^{N-1}(1+\log_2(w_{i,n})))$ $If\ LogRatio < MinLogRatio$
3	**SumOfPowers**	$N-1 \Big/ \sum_{n=1}^{N-1}((w_{i,n})^{Power})$
4	**PowerOfSums**	$N-1 \Big/ (w_{i,n})^{Power}$
5	**Mean**	$N-1 \Big/ w_{i,n}$
6	**NormSumOfLogs**	$\sum_{j=1}^{N-1} k_j \Big/ \sum_{n=1}^{N-1}(1+\log_2(w_{i,n}))$
7	**NormLogOfSums**	$\sum_{j=1}^{N-1} k_j \Big/ (1+\log_2(w_{i,n}))$
8	**NormSumOfPowers**	$\sum_{j=1}^{N-1} k_j \Big/ \sum_{n=1}^{N-1}(w_{i,n})^{Power}$
9	**NormSumsOfPowers**	$\frac{\sum_{j=1}^{N-1}(k_j)^{DocPower}}{\sum_{n=1}^{N-1}((w_{i,n})^{WordPower})}$
10	**SumOfLogs**	$N-1 \Big/ \sum_{n=1}^{N-1}(1+\log_2(w_{i,n}))$
11	**LogOfSums**	$N-1 \Big/ (1+\log_2(w_{i,n}))$
12	**NormMean**	$\sum_{j=1}^{N-1} k_j \Big/ w_{i,n}$
13	**NormPowerOfSums**	$\sum_{j=1}^{N-1} k_j \Big/ (w_{i,n})^{Power}$
14	**NormPowersOfSums**	$\frac{(\sum_{j=1}^{N-1} k_j)^{DocPower}}{(w_{i,n})^{WordPower}}$

Where:

i =current word
j = current document
k = total words in document *j*
n = total words in other than current document
N = total number of documents in the corpus
wi,j = number of occurrences of word *i* in document *j*.
wi,n = word occurrences of word *i* in other documents.
n_i = number of documents in which *i* occurs.
LogRatio = ratio of log for individual word to log for document length
MinLogRatio = user settable minimum for *LogRatio*
WordPower & *DocPower* = adjustable value, we used ***2*** in our experiments.

Measure: Mean_Attempts_to_Classify

As we report in [5], the Mean_Attempts_to_Classify measures how many attempts each TF*IDF equation classifies a file until it is classified correctly:

$$(\mathbf{1} \times \boldsymbol{P_1} + \mathbf{2} \times \boldsymbol{P_2} + \mathbf{3} \times \boldsymbol{P_3} \ldots + \mathbf{12} \times \boldsymbol{P_{12}}) \,/\, \boldsymbol{nfiles} \qquad (2)$$

Where:

$\boldsymbol{P_1} = \boldsymbol{number\ correctly\ classified\ on\ first\ try}$

$\boldsymbol{P_2} = \boldsymbol{number\ correctly\ classified\ after\ two\ tries}$

$\vdots$

$\boldsymbol{P_{12}} = \boldsymbol{number\ correctly\ classified\ on\ the\ last\ try}$

$\boldsymbol{nfiles} = \boldsymbol{number\ of\ files\ in\ testing\ class}$

In [5], we report on the mean attempts to classify ***all*** files in a class. This is an aggregate measure that can be quickly determined and used as a rough estimate as to which TF*IDF may be best suited for classifying a set a documents. However, here we report on a variation of the mean_attempts_to_classify in which we follow each and every file in the test set and determine how many misclassifications occur before the correct class is chosen.

One way to visualize mean_attempts_to_classify is by looking at the histogram of the number of tries each measure takes until the file is classified correctly. Figures 1a and 1b are examples. Here we show the attempts to classify for two different classes: *Business* and *Travel*. As Figure 1a shows, 18 of 49 *Business* class files are classified correctly by this TF*IDF on the first try. Figure 1c, however, shows that the same TF*IDF only classifies 3 files correctly on the first try for the *Travel* class. In fact, for *Travel*, the majority of the files (18) are never really correctly classified since the correct class is often chosen on the

12th try, only because there were no other classes to try. For comparison, Figure 1b shows the histogram for the same TF*IDF on the set of *Opinion* files.

We can use the results of these histograms to help us determine which TF*IDF might outperform other TF*IDF measures, and whether the classes themselves are too general to be used for classification purposes.

Measure: Attempt-Entropy

The entropy metric measures the amount of randomness in a system. For us, this means that as entropy increases for a specific TF*IDF measure, the classification results become more random. Equation 3 is used to determine the entropy:

$$\boldsymbol{e} = \sum_{i=1}^{n_{classes}} p_i \ln p_i \qquad (3)$$

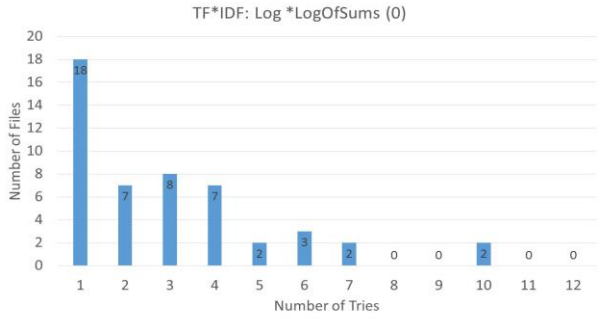

Figure 1a: Attempt_Histogram of TF*IDF for Business Class

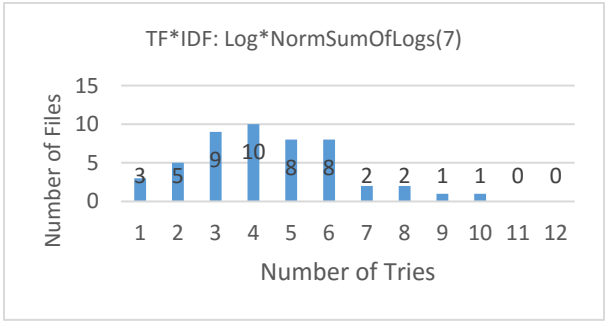

Figure 1b: Attempt_Histogram of TF*IDF for U.S. Class

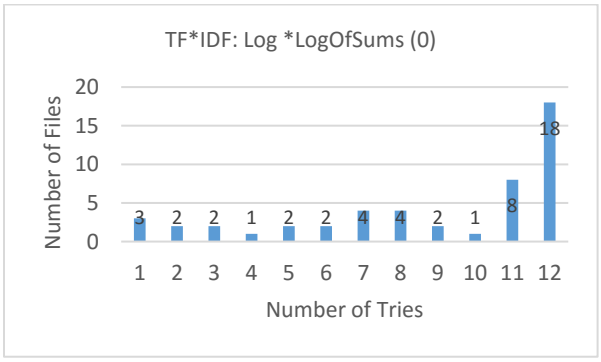

Figure 1c: Attempt_Histogram of TF*IDF for Travel Class

This measure is another method that can be used to help determine which TF*IDF metrics might be used in conjunction with each other to create more powerful classifiers than any single TF*IDF. Table 5 shows the attempt_entropy values for each of the first 25 TF*IDF measures for all classes.

Results

Table 3 shows a very small portion of the results for mean attempts for a single TF*IDF (Log*LogOfSums) and the results for the 1st four test files in the *Business* class. For each file, the maximum value indicates this particular TF*IDF 1st choice for class. In the example, the maximum value occurs for the *Business* class. This means that for TF*IDF# 0, the first file (file 0) is correctly classified on the first try. For the next two files in the example, the TF*IDF classifies both as *U.S.* (bolded values in the table) and correctly classifies these files on the second try. Finally, for the last example file, it takes this TF*IDF measure 3 tries to classify the file correctly.

Table 4 shows the 12 top performing TF*IDF measures using Mean_Attempts_to_Classify per file for each class. This can be compared to Table 7 which shows the top 12 performing TF*IDF based on accuracy as reported in [5]. Interestingly, the two different methods coincide in terms of the specific TF*IDF measures. While the ranking between the two lists are not the same, they only differ by two metrics. This strengthens the argument that most of these particular TF*IDF equations will produce the best results for the CNN articles.

The attempt-histograms help to determine the cohesiveness of a class. For example, Figure 1a shows a "left-skew" which tells us that this is a definite class, especially if most of the 112 TF*IDF have a similar skew for the set of test files. If, on the other hand, the histograms have a "right-skew" like that in Figure 1c, we can conclude this is not a class and the files either need a new class or to be re-assigned. Finally, if there is a "center skew" such as that seen in Figure 1b, this is may indicate that the class consists of two or more classes. The histograms themselves do not determine the cohesiveness of the class, for that we need also to look at attempt_entropy.

Table 5 shows the attempt_entropy results for the first 25 TF*IDF on each of the 12 classes. This allows for a side-by-side comparison of all the classes for the specific TF*IDF and allows us to get a sense for the differences between the classes. In this case, we cannot identify a *set* of best performing TF*IDFs for all classes. We can, however, look at individual class performance and identify the best TF*IDF metrics for each class. Table 6 shows this breakout and from this we can see that the top 12 performing TF*IDF measures for one class is not necessarily the same top 12 measures for another. This may be very useful when generating a list of TF*IDF metrics that may, in combination, generate better classification results than any single metric.

In general then, class cohesiveness can be represented when there is both low-entropy AND left-skew to the histogram.

Conclusions

We have presented two different methods for using the TF*IDF classification results reported in more depth in [5]. The *Mean_Attempts_to_Classification* can help to determine the optimal set of TF*IDF equations, similar to classification accuracy. *Attempt_Entropy*, on the other hand, may be useful for building class-specific classifiers. For example, we may be able to use a combined result of the best TF*IDF measures based on

attempt_entropy for the business class to create a *Business* classifier.

There are two next steps for this work. The first is to use the top performing TF*IDF measures to create "Meta-Classifiers" using meta-algorithmic patterns [12]. This should allow us to build classifiers that perform much better than any single TF*IDF. In addition, we have recently acquired the New York Time Annotated Corpus[13] and we are looking at running the current experiments on this very large corpus.

Table 3: TF*IDF values for 4 'Business' Class Files Illustrating Attempts to Classification = 1, 2 and 3.

TF * IDF #	File #	Business	Health	Justice	Living	Opinion	Politics	Showbiz	Sport	Tech	Travel	US	World
1	0	**1.1837**	0.6006	0.4362	0.4381	0.4713	0.4451	0.7506	0.4915	0.5571	0.3209	0.4396	0.6541
	1	0.6147	0.3969	0.4417	0.4311	0.4140	0.5448	0.4712	0.5280	0.6053	0.4229	**0.6300**	0.4548
	2	0.9938	0.9145	0.9489	0.8082	0.8166	0.7560	0.9004	0.9144	0.9288	0.8618	**1.1454**	0.8647
	3	1.0627	0.8928	1.0482	0.5378	0.8185	0.9517	**1.4141**	1.1977	0.9939	0.5098	0.7561	1.0587

Table 4: Mean Attempts by File: Top 12 Performing TF*IDF

TF * IDF	Busi-ness	Health	Justice	Living	Opin-ion	Politics	Show-biz	Sport	Tech	Travel	U.S.	World	Ave. Mean At-tempt
5	2	2	2.1633	4.9592	5.0408	2.4286	1.8776	1.8163	2.4898	6.1224	3.4082	2.2653	3.0476
3	1.898	1.8571	1.8776	6.0816	5.9184	2.0204	1.5714	1.6327	2.449	7.449	3.2857	2.0816	3.1769
11	2.2857	2.1633	2.449	4.7551	5.1224	2.4898	2.1224	1.9796	3	6.2041	3.5714	2.4082	3.2126
53	2.1837	2.1633	2.3878	4.8571	5.1429	2.5918	1.9388	2	2.9592	6.6735	3.5918	2.4082	3.2415
39	2.1837	2.1837	2.3878	4.8571	5.1429	2.5918	1.9388	2	2.9592	6.6735	3.5918	2.4082	3.2432
25	2.4286	2.449	2.4286	4.8571	5.0204	2.5918	2.2449	2.0816	3.1837	5.9184	3.6531	2.8776	3.3112
6	2.6939	2.5102	2.8571	4.1429	4.8163	3.3061	2.6531	2.4898	3.4898	5.3673	3.7959	3.0204	3.4286
1	2.6735	2.449	2.8776	4.1224	4.8163	3.2449	2.6531	2.5306	3.5102	5.5918	3.6531	3.0204	3.4286
101	2.7755	2.0612	2.551	5.5714	4.7959	2.6122	2.2245	1.8367	3.1837	6.8571	3.5714	3.1429	3.432
49	2.7551	2.449	2.7755	4.2653	5	3.2857	2.6122	2.5102	3.5102	5.7755	3.7143	2.9796	3.4694
35	2.7347	2.449	2.7755	4.3878	5.0612	3.2653	2.5918	2.4898	3.5102	6.0408	3.6939	2.9796	3.4983
2	2.8163	2.6531	1.7551	6.102	5.4898	2.5102	2.0612	2.3061	3.0204	7.8776	3.4082	2.7551	3.5629

Table 5: Attempt_Entropy by TF*IDF Numbers 1-25 for all Classes

TF*IDF Num	Business	Health	Justice	Living	Opinion	Politics	Showbiz	Sport	Tech	Travel	U.S	World
1	1.7824	1.7634	1.2279	2.2715	2.3181	1.6408	1.3688	1.2272	1.8734	2.0555	1.9356	1.7804
2	1.7279	1.6741	1.1375	2.3292	2.3174	1.6043	1.3471	1.2272	1.8108	2.2052	1.9416	1.6759
3	1.1641	1.1767	1.1849	2.3316	2.2199	1.3480	0.9175	0.9468	1.5979	2.1731	1.8988	1.3117
4	1.7191	2.0005	1.5711	2.0625	2.3708	2.0673	1.4350	1.8859	1.8783	1.9013	2.1208	1.6560
5	1.2239	1.3060	1.3470	2.2874	2.1118	1.5082	1.1352	1.1743	1.5929	2.2504	1.9657	1.4537
6	1.6847	1.4545	1.6453	2.0960	2.2926	1.9341	1.6776	1.4413	1.8904	2.2407	2.0796	1.8462
7	1.6872	1.4505	1.6453	2.1825	2.2851	1.9219	1.6776	1.4814	1.8841	2.2758	1.9966	1.7741
8	1.1942	1.9343	1.2755	1.6169	1.6498	1.6898	0.4956	1.1822	1.8624	0.9392	2.0513	1.2445
9	1.3786	1.9586	1.4488	1.8542	1.7914	1.9046	0.6148	1.2848	1.8969	1.1068	2.0790	1.3030
10	1.6981	2.1318	1.5768	1.3921	1.8545	2.0345	0.8872	1.7883	1.9765	1.2796	2.2163	1.5672
11	1.4603	1.3910	1.5173	2.2817	2.2117	1.5114	1.2836	1.2820	1.7693	2.2793	2.0274	1.4926
12	1.5629	2.2005	1.2621	1.1205	1.3435	1.9711	0.2976	1.4663	1.9549	0.7161	2.1514	1.5106
13	1.5890	2.1523	1.2066	0.8372	0.8257	1.9589	0.2693	1.3933	1.9906	0.4956	2.1170	1.4220
14	1.5665	2.2173	1.1882	0.9403	0.9204	1.9738	0.2693	1.4177	1.9782	0.6031	2.0717	1.4205
15	1.757	2.1939	1.3277	0.9263	1.0948	2.0138	0.3806	1.4967	2.0630	0.4687	2.1004	1.6320
16	1.7558	2.2071	1.3321	0.9985	1.2287	1.9292	0.3676	1.4862	2.0224	0.7496	2.1725	1.5560
17	1.4058	1.8500	1.2950	1.3950	1.7475	1.6744	0.3806	0.9522	1.8638	1.0842	2.0044	1.3201
18	1.7772	2.1370	1.6356	1.7531	2.0905	2.1193	1.1753	1.9137	1.9367	1.4323	2.2172	1.6294
19	1.2763	1.9482	1.2663	1.7790	1.9501	1.8146	0.6607	1.1415	1.8991	1.4623	2.0910	1.4905
20	1.6283	1.9209	1.6471	2.2895	2.2473	1.8920	1.1003	1.4845	2.0643	2.1026	2.1228	1.7456
21	1.7134	1.7295	1.7315	2.2153	2.2574	1.9441	1.8047	1.6788	1.9369	2.2036	2.0447	1.9336
22	1.2445	1.9295	1.3152	1.8147	1.9695	1.8255	0.6607	1.1162	1.8801	1.4500	2.0940	1.5371
23	1.4429	1.9765	1.4001	1.8391	2.0267	1.8118	0.6431	1.3052	1.9735	1.5321	2.1495	1.4905
24	1.7772	2.1490	1.6356	1.7531	2.0394	2.1193	1.1580	1.9137	1.9311	1.3754	2.2293	1.6753
25	1.5494	1.5280	1.5422	2.2784	2.1835	1.5610	1.3677	1.3213	1.8535	2.2781	2.0236	1.7942

Table 6: Entropy by Class – Top 12 performing TF*IDF

Business

TF*IDF Number	Entropy
[3]	1.1641
[8]	1.1942
[5]	1.2239
[22]	1.2445
[36]	1.2601
[33]	1.2642
[50]	1.2676
[19]	1.2763
[47]	1.2978
[31]	1.3098
[45]	1.3275
[9]	1.3786

Health

TF*IDF Number	Entropy
[3]	1.8571
[5]	2
[101]	2.0612
[11]	2.1633
[53]	2.1633
[39]	2.1837
[7]	2.449
[25]	2.449
[35]	2.449
[49]	2.449
[6]	2.5102
[2]	2.6531

Justice

TF*IDF Number	Entropy
[58]	0.9693
[57]	0.9716
[69]	0.9875
[30]	1.0152
[44]	1.0594
[42]	1.077
[56]	1.077
[70]	1.077
[41]	1.0979
[55]	1.0979
[68]	1.0992
[29]	1.108

Living

TF*IDF Number	Entropy
[29]	0.7231
[43]	0.7231
[57]	0.7437
[30]	0.7572
[58]	0.814
[70]	0.814
[13]	0.8372
[41]	0.8372
[44]	0.8372
[55]	0.8372
[69]	0.8903
[42]	0.9041

Opinion

TF*IDF Number	Entropy
[29]	0.6431
[43]	0.6431
[31]	0.689
[30]	0.7163
[41]	0.7928
[44]	0.7928
[55]	0.7928
[13]	0.8257
[57]	0.8445
[59]	0.8855
[14]	0.9204
[42]	0.9455

Politics

TF*IDF Number	Entropy
[3]	1.348
[5]	1.5082
[11]	1.5114
[25]	1.561
[73]	1.5644
[87]	1.5644
[39]	1.5945
[53]	1.5945
[2]	1.6043
[101]	1.6401
[1]	1.6408
[17]	1.6744

Showbiz

TF*IDF Number	Entropy
[31]	0
[45]	0
[59]	0.0996

Sport

TF*IDF Number	Entropy
[3]	0.9468
[17]	0.9522
[59]	1.086

Tech

TF*IDF Number	Entropy
[5]	1.5929
[3]	1.5979
[39]	1.7218

[61]	0.1705	[61]	1.086	[53]	1.7218
[69]	0.1988	[64]	1.086	[11]	1.7693
[64]	0.2303	[45]	1.1141	[73]	1.8045
[13]	0.2693	[22]	1.1162	[87]	1.8045
[14]	0.2693	[31]	1.117	[2]	1.8108
[29]	0.2693	[19]	1.1415	[101]	1.8168
[30]	0.2693	[101]	1.1559	[106]	1.8212
[40]	0.2693	[5]	1.1743	[103]	1.8407
[41]	0.2693	[8]	1.1822	[33]	1.8469

Travel		**U.S.**		**World**	
TF*IDF Number	**Entropy**	**TF*IDF Number**	**Entropy**	**TF*IDF Number**	**Entropy**
[29]	0.2693	[106]	1.853	[45]	1.2414
[31]	0.3718	[3]	1.8988	[8]	1.2445
[59]	0.3806	[1]	1.9356	[31]	1.2655
[30]	0.3982	[2]	1.9416	[37]	1.2933
[41]	0.3982	[103]	1.9424	[51]	1.2933
[43]	0.3982	[101]	1.9612	[9]	1.303
[55]	0.3982	[5]	1.9657	[33]	1.3035
[57]	0.3982	[31]	1.9687	[47]	1.3035
[58]	0.3982	[61]	1.9752	[50]	1.3099
[69]	0.3982	[45]	1.9839	[3]	1.3117
[44]	0.4497	[36]	1.9886	[17]	1.3201
[15]	0.4687	[49]	1.9888	[36]	1.3206

References

[1] Levinson, S. Pragmatics. Cambridge University Press, New York, NY. 1983

[2] I. Gil-Leiva, & A. Alonso-Arroyo. Keywords given by authors of scientific articles in database descriptors. Journal of the American society for information science and technology, 58(8), (2007)1175-1187

[3] R.D. Lins, S.J. Simske, L. Cabral, G. Silva, R. Lima, R.F. Mello, and L. Favaro. A multi-tool scheme for summarizing textual documents. In Proceedings of 11st IADIS International Conference WWW/INTERNET 2012. pp. 1–8.

[4] K. K. Bun, and I. Mitsuru. "Topic Extraction from News Archive Using TF* PDF Algorithm." WISE. 2002.

[5] A. M.Vans and S. J. Simske, Identifying top performing TF*IDF classifiers using the CNN corpus. Submitted for publication to the Journal of Imaging Science and Technology (JIST), September, 2016.

[6] Gerard Salton and Christopher Buckley, Term-Weighting Approaches in Automatic Text Retrieval, Information Processing and Management 24.5 (1988): 513-23.

[7] Stephen Robertson, Understanding inverse document frequency: on theoretical arguments for IDF, Journal of documentation 60.5 (2004): 503-520.

[8] K.L Kwok, Experiments with a component theory of probabilistic informational retrieval based on single terms as document components. ACM Transactions on Information Systems, 8(4). (1990), Pp. 363-386.

[9] J. Ramos, Using tf-idf to determine word relevance in document queries, Proceedings of the first instructional conference on machine learning (2003).

[10] S. Karbasi, and M Boughanem, Effective level of term frequency impact on large-scale retrieval performance: by top-term ranking method, Proceedings of the 1st international conference on Scalable information systems, ACM (2006), pp, 37.

[11] CodePlex. 2013. SharpNLP – open source natural language processing tools. Retrieved from https://sharpnlp.codeplex.com/#.

[12] S. J. Simske, Meta-algorithmics: patterns for robust, low cost, high quality systems. John Wiley & Sons, 2013.

[13] New York Times Annotated Corpus, Linguistic Data Consortium (LDC), https://catalog.ldc.upenn.edu/LDC2008T19, Accessed September 13, 2016.

Author Biography

Steven Simske is an HP Fellow and a Director in HP Labs. He is the author of more than 400 publications and roughly 140 US patents. He is an IS&T Fellow and an honorary professor at the University of Nottingham. Steve has been a member of the World Economic Forum Global Agenda Councils since 2010, including Illicit Trade, Illicit Economy and the Future of Electronics. At HP, he directs teams in research on 3D printing, education, life sciences, sensing, authentication, packaging, imaging and manufacturing. His book "Meta-Algorithmics" addresses intelligent systems.

Table 7: Top 12 performing TF*IDF – 2nd Column Data from [5].

Top TF*IDF Ranked Based on Accuracy	Accuracy
TF_Log_IDF_NormLogOfSums (3)	0.447
TF_Log_IDF_NormMean (5)	0.439
TF_NormLogs_IDF_NormSumsOfPowers(53)	0.410
TF_NormLog_IDF_NormSumsOfPowers (39)	0.408
TF_Log_IDF_NormSumsOfPowers (11)	0.405
TF_Log_IDF_NormPowerOfSums (6)	0.364
TF_Log_IDF_NormPowersOfSums (7)	0.362
TF_Mean_IDF_NormSumsOfPowers (25)	0.362
TF_NormLogs_IDF_NormPowersOfSums (49)	0.362
TF_NormLog_IDF_NormPowersOfSums (35)	0.359
TF_Log_IDF_Mean (2)	0.359
TF_Log_IDF_LogOfSums (1)	0.340

Top 12 TF*IDF Ranked Based on Mean Attempts	Average Mean Attempts
TF_Log_IDF_NormMean (5)	3.048
TF_Log_IDF_NormLogOfSums (3)	3.177
TF_Log_IDF_NormSumsOfPowers (11)	3.213
TF_NormLogs_IDF_NormSumsOfPowers (53)	3.242
TF_NormLog_IDF_NormSumsOfPowers (39)	3.243
TF_Mean_IDF_NormSumsOfPowers (25)	3.311
TF_Log_IDF_NormPowerOfSums (6)	3.429
TF_Log_IDF_LogOfSums (1)	3.429
TF_Power_IDF_NormLogOfSums (101)	3.432
TF_NormLogs_IDF_NormPowersOfSums (49)	3.469
TF_NormLog_IDF_NormPowersOfSums (35)	3.498
TF_Log_IDF_Mean (2)	3.563