DOI: 10.2352/issn.2168-3204.2016.1.0.21

Summarization and Classification of CNN.com Articles using the TF*IDF Family of Metrics

Marie Vans & Steven Simske; HP Labs; Fort Collins, CO

Abstract

*TF*IDF (term frequency times inverse document frequency) is a common metric used to automatically discover keywords in documents for use in classification and other text processing applications. We are interested in determining whether these measures can help in determining the most relevant sentences for summarization and classification purposes. However, there are many ways to define TF*IDF, and to date no attempt to relatively—and systematically—gauge the value of these different forms has been performed. We investigate a comprehensive family of 112 TF*IDF measures (corresponding to an a priori estimate of 20 degrees of freedom among these measures) applied to 3000 CNN articles belonging in 12 classes such as Business, Sport, and World. The assumption is that at least some sets of these measures must be effective for document summarization and classification. The goal is to identify the summaries provided by TF*IDF measures that best match human generated summaries as well as find effective TF*IDF definitions for classification purposes.*

.Introduction

The general principle of using relatively rare terms to identify a document extends back several decades. [1]. One popular method of identifying these relatively rare terms is the term frequency times the inverse of the document frequency, or TF*IDF. The TF*IDF measurement increases for a given document in proportion to the specific term's occurrence therein, and inversely proportionate to its occurrence in other documents. All of which make sense – in the limit a term, such as a neologism, that only occurs in one document is a perfect query or indexing term for that document. However, TF*IDF has at least 112 mathematical formulations, and to date no one has determined which is optimal for a given corpus, or how much the performance of these different incarnations of the TF*IDF varies.

We aim to change this in this paper. We define a family of 112 TF*IDF measures and consider their utility for summarization and classification of a ground-truthed set of documents organized into 12 classes. This comprehensive study is used to identify which paired combinations of 8 different TF and 14 different IDF formulations (meaning there are 7 degrees of freedom for TF and 13 degrees of freedom for IDF) are likely to be useful for these two distinct archiving tasks (summarization and classification). In so doing, we provide insight into the nature of term rareness for the automation of corpus tagging and usability throughout its life cycle.

TF*IDF

Automatic keyword extraction is a common goal for information retrieval and query formation. Many have run experiments using less structured or diverse document collections such as emails and webpages [2, 3, 4]. Of the many approaches available, TF*IDF (Term Frequency x Inverse Document Frequency) [5, 6, 7, 8] is commonly used in information retrieval and classification tasks [9, 10, 11]. As mentioned above, 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 3000 CNN articles. For this paper, we focus on determining the effectiveness of using the set of TF*IDF measures on all sentences in the CNN file and then, separately, the ground-truthed Gold Standard sentences. Table 1 shows a breakdown of the 8 term frequency (TF) measures while Table 2 shows 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.

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

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 [12] C# open source project. The stream is then converted into a bag of words consisting of stemmed words from each sentence with stop words removed. We then calculate a measure for each sentence by summing the values of TF*IDF measures for each stemmed word found in the sentence.

Table 1: TF Equations Used in Experiments

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

Table 2: IDF Equations Used in Experiments.

	IDF Name	IDF Denominator

1	**NormLogsOfSums**	$\frac{\log_{\frac{2}{LogRatio}}\left(\sum_{j=1}^{N-1} k_j\right)}{1+\log_2\left(w_{i,n}\right)}$
		if LogRatio ≥ MinLogRatio
		$\log_2\left(\sum_{j=1}^{N-1} k_j\right) / 1+\log_2\left(w_{i,n}\right)$
		if LogRatio < MinLogRatio

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

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.

In this paper, we will report on summarization and classification results using these 112 TF*IDF with an emphasis on identifying which combinations of TF and IDF formulations (of 8 and 14, respectively) might be the most useful for these two distinct archiving tasks (summarization and classification). Using the relative difference of these measures, and their overall effectiveness, we are looking to provide insight into the nature of term rareness for the automation of corpus tagging and corpus usability throughout its life cycle.

References

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

[2] Turney, P. D. (2000). Learning algorithms for keyphrase extraction. *Information Retrieval*, *2*(4), 303-336.

[3] Turney, P. (2002). Mining the Web for Lexical Knowledge to Improve Keyphrase Extraction: Learning from Labeled and Unlabeled Data.

[4] El-Beltagy, S. R. (2006, November). KP-Miner: A Simple System for Effective Keyphrase Extraction. In *Innovations in Information Technology, 2006* (pp. 1-5). IEEE.

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

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

[7] Manning, C. and Schütze, H. 1999. Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA.

[8] Papineni, Kishore. "Why inverse document frequency?." Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies. Association for Computational Linguistics, 2001.

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

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

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

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

Author Biography

DOI: 10.2352/issn.2168-3204.2016.1.0.21

Marie Vans is currently a Research Scientist with Hewlett-Packard Labs in Fort Collins, Colorado. Her main interests are security printing and imaging for document workflows, statistical language processing, and other approaches to document understanding. She holds a Ph.D. in Computer Science from Colorado State University.