tf–idf

From Wikipedia, the free encyclopedia

In information retrieval, **tf–idf**, short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.[1] It is often used as a weighting factor in information retrieval, text mining, and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes. For instance, 83% of text-based recommender systems in the domain of digital libraries use tf-idf.[2]

Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

One of the simplest ranking functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model.

Contents

Motivations

Term frequency

Suppose we have a set of English text documents and wish to determine which document is most relevant to the query "the brown cow". A simple way to start out is by eliminating documents that do not contain all three words "the", "brown", and "cow", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its *term frequency*. However, in the case where the length of documents vary greatly, adjustments are often made (see definition below).

The first form of term weighting is due to Hans Peter Luhn (1957) and is based on the Luhn Assumption:

- The weight of a term that occurs in a document is simply proportional to the term frequency.[3]

Inverse document frequency

Because the term "the" is so common, term frequency will tend to incorrectly emphasize documents which happen to use the word "the" more frequently, without giving enough weight to the more meaningful terms "brown" and "cow". The term "the" is not a good keyword to distinguish relevant and non-relevant documents and terms, unlike the less common words "brown" and "cow". Hence an *inverse document frequency* factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely.

Karen Spärck Jones (1972) conceived a statistical interpretation of term specificity called Inverse Document Frequency (IDF), which became a cornerstone of term weighting:

- The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.[4]

Definition

tf–idf is the product of two statistics, term frequency and inverse document frequency. Various ways for determining the exact values of both statistics exist.

Term frequency

Variants of term frequency (TF) weight

weighting scheme	**TF weight**
binary	$0, 1$
raw count	$f_{t,d}$
term frequency	$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$
log normalization	$1 + \log(f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

In the case of the **term frequency** $\mathrm{tf}(t,d)$, the simplest choice is to use the *raw count* of a term in a document, i.e. the number of times that term t occurs in document d. If we denote the raw count by $f_{t,d}$, then the simplest tf scheme is $\mathrm{tf}(t,d) = f_{t,d}$. Other possibilities include[5]:128

- Boolean "frequencies": $\mathrm{tf}(t,d) = 1$ if t occurs in d and 0 otherwise;
- term frequency adjusted for document length : $f_{t,d} \div$ (number of words in d)
- logarithmically scaled frequency: $\mathrm{tf}(t,d) = 1 + \log f_{t,d}$, or zero if $f_{t,d}$ is zero;
- augmented frequency, to prevent a bias towards longer documents, e.g. raw frequency divided by the raw frequency of the most occurring term in the document:

$$\mathrm{tf}(t,d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}}$$

Inverse document frequency

The **inverse document frequency** is a measure of how much information the word provides, that is, whether the term is common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

Variants of inverse document frequency (IDF) weight

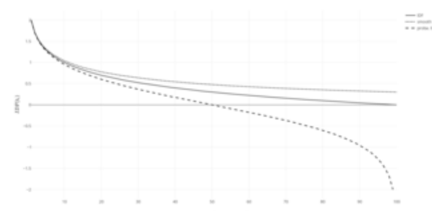
weighting scheme	IDF weight ($n_t = \|\{d \in D : t \in d\}\|$)
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log \left(1 + \frac{N}{n_t}\right)$
inverse document frequency max	$\log \left(\frac{\max_{\{t' \in d\}} n_{t'}}{1 + n_t}\right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

$$\mathrm{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

with

- N: total number of documents in the corpus $N = |D|$
- $|\{d \in D : t \in d\}|$: number of documents where the term t appears (i.e., $\mathrm{tf}(t, d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{d \in D : t \in d\}|$.

Term frequency–Inverse document frequency



Plot of differents inverse document frequency functions: standard, smooth, probabilistic.

Then tf–idf is calculated as

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t, D)$$

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to filter out common terms. Since the ratio inside the idf's log function is always greater than or equal to 1, the value of idf (and tf-idf) is greater than or equal to 0. As a term appears in more documents, the ratio inside the logarithm approaches 1, bringing the idf and tf-idf closer to 0.

Recommended TF-IDF weighting schemes

weighting scheme	document term weight	query term weight
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5 \frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
2	$1 + \log f_{t,d}$	$\log(1 + \frac{N}{n_t})$
3	$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$

Justification of idf

Idf was introduced, as "term specificity", by Karen Spärck Jones in a 1972 paper. Although it has worked well as a heuristic, its theoretical foundations have been troublesome for at least three decades afterward, with many researchers trying to find information theoretic justifications for it.[6]

Spärck Jones's own explanation did not propose much theory, aside from a connection to Zipf's law.[6] Attempts have been made to put idf on a probabilistic footing,[7] by estimating the probability that a given document d contains a term t as the relative document frequency,

$$P(t|d) = \frac{|\{d \in D : t \in d\}|}{N},$$

so that we can define idf as

$$\begin{aligned} \mathrm{idf} &= -\log P(t|d) \\ &= \log \frac{1}{P(t|d)} \\ &= \log \frac{N}{|\{d \in D : t \in d\}|} \end{aligned}$$

Namely, the inverse document frequency is the logarithm of "inverse" relative document frequency.

This probabilistic interpretation in turn takes the same form as that of self-information. However, applying such information-theoretic notions to problems in information retrieval leads to problems when trying to define the appropriate event spaces for the required probability distributions: not only documents need to be taken into account, but also queries and terms.[6]

Example of tf–idf

Suppose that we have term count tables of a corpus consisting of only two documents, as listed on the right.

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
is	1
another	2
example	3

The calculation of tf–idf for the term "this" is performed as follows:

In its raw frequency form, tf is just the frequency of the "this" for each document. In each document, the word "this" appears once; but as the document 2 has more words, its relative frequency is smaller.

$$\begin{aligned} \mathrm{tf}(''\mathsf{this}'', d_1) &= \frac{1}{5} = 0.2 \\ \mathrm{tf}(''\mathsf{this}'', d_2) &= \frac{1}{7} \approx 0.14 \end{aligned}$$

An idf is constant per corpus, and **accounts** for the ratio of documents that include the word "this". In this case, we have a corpus of two documents and all of them include the word "this".

$$\mathrm{idf}(''\mathsf{this}'', D) = \log\left(\frac{2}{2}\right) = 0$$

So tf–idf is zero for the word "this", which implies that the word is not very informative as it appears in all documents.

$$\mathrm{tfidf}(''\mathsf{this}'', d_1) = 0.2 \times 0 = 0$$
$$\mathrm{tfidf}(''\mathsf{this}'', d_2) = 0.14 \times 0 = 0$$

A slightly more interesting example arises from the word "example", which occurs three times but only in the second document:

$$\mathrm{tf}(''\mathsf{example}'', d_1) = \frac{0}{5} = 0$$
$$\mathrm{tf}(''\mathsf{example}'', d_2) = \frac{3}{7} \approx 0.429$$
$$\mathrm{idf}(''\mathsf{example}'', D) = \log\left(\frac{2}{1}\right) = 0.301$$

Finally,

$$\mathrm{tfidf}(''\mathsf{example}'', d_1) = \mathrm{tf}(''\mathsf{example}'', d_1) \times \mathrm{idf}(''\mathsf{example}'', D) = 0 \times 0.301 = 0$$
$$\mathrm{tfidf}(''\mathsf{example}'', d_2) = \mathrm{tf}(''\mathsf{example}'', d_2) \times \mathrm{idf}(''\mathsf{example}'', D) = 0.429 \times 0.301 \approx 0.13$$

(using the base 10 logarithm).

tf-idf Beyond Terms

The idea behind TF–IDF has also been applied to entities other than terms. In 1998, the concept of IDF was applied to citations.[8] The authors argued that "if a very uncommon citation is shared by two documents, this should be weighted more highly than a citation made by a large number of documents". In addition, tf-idf was applied to "visual words" with the purpose of conducting object matching in videos,[9] and entire sentences.[10] However, not in all cases did the concept of TF–IDF proved to be more effective than a plain TF scheme (without IDF). When TF–IDF was applied to citations, researchers could find no improvement over a simple citation–count weight that had no IDF component.[11]

tf-idf Derivates

There are a number of term-weighting schemes that derived from TF–IDF. One of them is TF–PDF (Term Frequency * Proportional Document Frequency).[12] TF-PDF was introduced in 2001 in the context of identifying emerging topics in the media. The PDF component measures the difference of how often a term occurs in different domains. Another derivate is TF-IDuF. In TF-IDuF,[13] IDF is not calculated based on the document corpus that is to be searched or recommended. Instead, IDF is calculated based on users' personal document collections. The authors report that TF-IDuF was equally effective as tf-idf but could also be applied in situations when e.g. a user modeling system has no access to a global document corpus.

See also

- Okapi BM25
- Noun phrase
- Word count
- Vector space model
- PageRank
- Kullback–Leibler divergence
- Mutual information
- Latent semantic analysis

- Latent semantic indexing
- Latent Dirichlet allocation

References

1. Rajaraman, A.; Ullman, J. D. (2011). "Data Mining". *Mining of Massive Datasets* (http://i.stanford.edu/~ullman/mmds/ch1.pdf) (PDF). pp. 1–17. ISBN 978-1-139-05845-2. doi:10.1017/CBO9781139058452.002 (https://doi.org/10.1017%2FCBO9781139058452.002).
2. Breitinger, Corinna; Gipp, Bela; Langer, Stefan (2015-07-26). "Research-paper recommender systems: a literature survey" (http://link.springer.com/article/10.1007/s00799-015-0156-0). *International Journal on Digital Libraries*. **17** (4): 305–338. ISSN 1432-5012 (https://www.worldcat.org/issn/1432-5012). doi:10.1007/s00799-015-0156-0 (https://doi.org/10.1007%2Fs00799-015-0156-0).
3. Luhn, Hans Peter (1957). "A Statistical Approach to Mechanized Encoding and Searching of Literary Information" (http://web.stanford.edu/class/linguist289/luhn57.pdf) (PDF). *IBM Journal of research and development*. IBM. **1** (4): 315. doi:10.1147/rd.14.0309 (https://doi.org/10.1147%2Frd.14.0309). Retrieved 2 March 2015. "There is also the probability that the more frequently a notion and combination of notions occur, the more importance the author attaches to them as reflecting the essence of his overall idea."
4. Spärck Jones, K. (1972). "A Statistical Interpretation of Term Specificity and Its Application in Retrieval" (http://www.emeraldinsight.com/doi/abs/10.1108/eb026526). *Journal of Documentation*. **28**: 11–21. doi:10.1108/eb026526 (https://doi.org/10.1108%2Feb026526).
5. Manning, C. D.; Raghavan, P.; Schutze, H. (2008). "Scoring, term weighting, and the vector space model". *Introduction to Information Retrieval* (http://nlp.stanford.edu/IR-book/pdf/06vect.pdf) (PDF). p. 100. ISBN 978-0-511-80907-1. doi:10.1017/CBO9780511809071.007 (https://doi.org/10.1017%2FCBO9780511809071.007).
6. Robertson, S. (2004). "Understanding inverse document frequency: On theoretical arguments for IDF". *Journal of Documentation*. **60** (5): 503–520. doi:10.1108/00220410410560582 (https://doi.org/10.1108%2F00220410410560582).
7. See also Probability estimates in practice (http://nlp.stanford.edu/IR-book/html/htmledition/probability-estimates-in-practice-1.html#p:justificationofidf) in *Introduction to Information Retrieval*.
8. Bollacker, Kurt D.; Lawrence, Steve; Giles, C. Lee (1998-01-01). "CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications" (http://doi.acm.org/10.1145/280765.280786). *Proceedings of the Second International Conference on Autonomous Agents*. AGENTS '98. New York, NY, USA: ACM: 116–123. ISBN 0-89791-983-1. doi:10.1145/280765.280786 (https://doi.org/10.1145%2F280765.280786).
9. Sivic, Josef; Zisserman, Andrew (2003-01-01). "Video Google: A Text Retrieval Approach to Object Matching in Videos" (http://dl.acm.org/citation.cfm?id=946247.946751). *Proceedings of the Ninth IEEE International Conference on Computer Vision – Volume 2*. ICCV '03. Washington, DC, USA: IEEE Computer Society: 1470–. ISBN 0-7695-1950-4.
10. Seki, Yohei. "Sentence Extraction by tf/idf and Position Weighting from Newspaper Articles" (http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings3/NTCIR3-TSC-SekiY.pdf) (PDF). National Institute of Informatics.
11. Beel, Joeran; Breitinger, Corinna (2017). "Evaluating the CC-IDF citation-weighting scheme – How effectively can 'Inverse Document Frequency' (IDF) be applied to references?" (http://beel.org/publications/2017%20iConference%20--%20Evaluating%20the%20CC-IDF%20citation-weighting%20scheme%20--%20preprint.pdf) (PDF). *Proceedings of the 12th iConference*.
12. Khoo Khyou Bun; Bun, Khoo Khyou; Ishizuka, M. (2001). "Emerging Topic Tracking System" (https://doi.org/10.1109/WECWIS.2001.933900). *Proceedings Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems. WECWIS 2001*: 2. ISBN 0-7695-1224-0. doi:10.1109/wecwis.2001.933900 (https://doi.org/10.1109%2Fwecwis.2001.933900).

13. Langer, Stefan; Gipp, Bela (2017). "TF-IDuF: A Novel Term-Weighting Scheme for User Modeling based on Users' Personal Document Collections" (https://www.gipp.com/wp-content/papercite-data/pdf/beel17.pdf) (PDF). *iConference*.

- Salton, G; McGill, M. J. (1986). *Introduction to modern information retrieval*. McGraw-Hill. ISBN 978-0-07-054484-0.
- Salton, G.; Fox, E. A.; Wu, H. (1983). "Extended Boolean information retrieval". *Communications of the ACM*. **26** (11): 1022–1036. doi:10.1145/182.358466 (https://doi.org/10.1145%2F182.358466).
- Salton, G.; Buckley, C. (1988). "Term-weighting approaches in automatic text retrieval". *Information Processing & Management*. **24** (5): 513–523. doi:10.1016/0306-4573(88)90021-0 (https://doi.org/10.1016%2F0306-4573%2888%2990021-0).
- Wu, H. C.; Luk, R. W. P.; Wong, K. F.; Kwok, K. L. (2008). "Interpreting TF-IDF term weights as making relevance decisions". *ACM Transactions on Information Systems*. **26** (3): 1. doi:10.1145/1361684.1361686 (https://doi.org/10.1145%2F1361684.1361686).

External links and suggested reading

- TFxIDF Repository (http://kak.tx0.org/Information-Retrieval/TFxIDF): A definitive guide to the variants and their evolution.
- Gensim is a Python library for vector space modeling and includes tf–idf weighting.
- Robust Hyperlinking (http://bscit.berkeley.edu/cgi-bin/pl_dochome?query_src=&format=html&collection=Wilensky_papers&id=3&show_doc=yes): An application of tf–idf for stable document addressability.
- A demo of using tf–idf with PHP and Euclidean distance for Classification (http://infinova.wordpress.com/2010/01/26/distance-between-documents/)
- Anatomy of a search engine (http://www.codeproject.com/KB/IP/AnatomyOfASearchEngine1.aspx)
- tf–idf and related definitions (http://lucene.apache.org/core/3_6_1/api/all/org/apache/lucene/search/Similarity.html) as used in Lucene
- TfidfTransformer (http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html#sklearn.feature_extraction.text.TfidfTransformer) in scikit-learn
- Text to Matrix Generator (TMG) (http://scgroup.hpclab.ceid.upatras.gr/scgroup/Projects/TMG/) MATLAB toolbox that can be used for various tasks in text mining (TM) specifically i) indexing, ii) retrieval, iii) dimensionality reduction, iv) clustering, v) classification. The indexing step offers the user the ability to apply local and global weighting methods, including tf–idf.
- Pyevolve: A tutorial series explaining the tf-idf calculation (http://blog.christianperone.com/?p=1589).
- TF/IDF with Google n-Grams and POS Tags (http://trimc-nlp.blogspot.com/2013/04/tfidf-with-google-n-grams-and-pos-tags.html)