N-gram IDF: A Global Term Weighting Scheme Based on Information Distance

Masumi Shirakawa, Takahiro Hara, Shojiro Nishio
Osaka University, JAPAN

24th International World Wide Web Conference (WWW 2015)
May 18–22, 2015, Florence, Italy.
Contributions

1. Give a new explanation of IDF.

2. Propose a new IDF scheme that can handle N-grams of any N.

3. Propose an implementation of 2.

4. Exemplify the potential of 2.
Inverse Document Frequency (IDF)

Give more weight to a term occurring in less documents

$$IDF(t) = \log \frac{|D|}{df(t)}$$

- $t$: Term
- $df(t)$: Document frequency of $t$
- $|D|$: Number of documents in $D$

"algorithm"  
IDF is large

"you"  
IDF is small
Weak point of IDF

IDF does not work well for N-grams (phrases).

**WHY?**

N-gram occurring in less documents is more likely to be a key term. N-gram of unnatural collocation occurs in less documents.

N-gram of unnatural collocation is more likely to be a key term.

Estimated DF using Web Search

\[ df("Leonardo da Vinci") = 31,700,000 \]
\[ df("Leonardo da is") = 15 \]

The definition of IDF totally contradicts the definition of good phrases.
Multiword Expression (MWE)

MWE is a major research topic in Natural Language Processing (NLP). IDF has been developed in Information Retrieval (IR).

Representative measures of MWE:

- PMI
- SCP [Silva+, MOL99]
- EMI [Zhang+, Expert Systems with Applications, 2009]
- MED [Bu+, COLING10]
- ...

There was no theoretical explanation to connect term weighting with MWE we have done.
Key Theories

Kolmogorov Complexity and Information Distance
Kolmogorov Complexity (1/2)

[Kolmogorov, Sankhya, 63]

Measure of the randomness of a (bit) string

$K(x)$: Kolmogorov complexity of $x$

Q₁: Which one has larger complexity?

$x₁ = “01010101010101010” \rightarrow “01” \times 7 + ”0”$

$x₂ = “011101100010110” \rightarrow “011101100010110”$

A₁: Probably $K(x₁) < K(x₂)$
Kolmogorov Complexity \((2/2)\)

[Kolmogorov, Sankhya, 63]

Measure of the randomness of a (bit) string

\(K(x|y)\): conditional Kolmogorov complexity of \(x\) given \(y\)

Q\(_2\): Which one has larger complexity? \((y\) can be used\)
\(y = \text{"01110110001011"}\)
\(x_1 = \text{"0101010101010101"} \rightarrow \text{"01" \times 7 + "0"}\)
\(x_2 = \text{"0111011000101101"} \rightarrow y + \text{"0"}\)

A\(_2\): Probably \(K(x_1|y) > K(x_2|y)\)

\(K(x, y) = K(x|y) + K(y) = K(y|x) + K(x)\)

(One string can be reused to describe the other)
Information Distance
[Bennett+, IEEE ToIT, 98]

Universal distance defined by Kolmogorov complexity

\[ E(x, y): \text{information distance between } x \text{ and } y \]
\[ E(x, y) = \max(K(x|y), K(y|x)) = K(x, y) - \min(K(y), K(x)) \]

It is equal to energy to convert one string to the other.

Energy cost to convert one bit = \(1kT \ln(2)\)
[Landauer, IBM Journal of Research and Development, 61]

Absolute temperature in Kelvin
Boltzmann constant
Indispensable Work

Multiword Expression Distance
Multiword Expression Distance (MED)
[Bu+, COLING10]

Measure of MWE based on information distance

\[ MED(g) = \log \frac{df(w_1, \ldots, w_N)}{df(g)} \]

Information distance between context and semantic

Context of \( g \): set of documents containing \( g \)
Semantic of \( g \): set of documents containing \( w_1, \ldots, w_N \)

Inspired by Normalized Google Distance [Cilibrasi+, TKDE, 2007]
Derivation of MED (1/2)

We assume that the probability of context $x$ is proportional to its cardinality $|x|$.

$$P(x) = \frac{|x|}{\sum_{x_i \in X} |x_i|}$$

$X$: set of all contexts

Then we can approximate the Kolmogorov complexity using Shannon–Fano coding*1. [Li+2008]

$$K(x) \approx - \log P(x)$$
$$K(x, y) \approx - \log P(x, y)$$

Derivation of MED (2/2)

Information distance between $\phi(g)$ and $\mu(g)$ is

\[
E(\phi(g), \mu(g)) = K(\phi(g), \mu(g)) - \min\left(K(\phi(g)), K(\mu(g))\right)
\]

\[
= -\log P(\phi(g), \mu(g)) + \min(\log P(\phi(g)), \log P(\mu(g)))
\]

\[
= -\log|\phi(g) \cap \mu(g)| + \max(\log|\phi(g)|, \log|\mu(g)|)
\]

\[
= -\log|\phi(g)| + \log|\mu(g)|, \quad \phi(g) \subseteq \mu(g)
\]

Finally we have

\[
MED(g) = \log \frac{df(w_1, \ldots, w_N)}{df(g)}
\]
IDF and information distance

**IDF**

\[
IDF(g) = \log \frac{|D|}{df(g)}
\]

Information distance between contexts of \( g \) and empty string \( \varepsilon \)

\[E(\phi(g), \phi(\varepsilon)) = -\log |\phi(g) \cap \phi(\varepsilon)| + \max(\log |\phi(g)|, \log |\phi(\varepsilon)|)\]

\[= -\log |\phi(g)| + \log |\phi(\varepsilon)| \quad \phi(g) \subseteq \phi(\varepsilon)\]

\[= \log \frac{df(\varepsilon)}{df(g)} = \log \frac{|D|}{df(g)} = IDF(g)\]
Contributions

1. Give a new explanation of IDF.
   
   IDF is equal to the distance between the term and the empty string in the information distance space.

2. Propose a new IDF scheme that can handle N-grams of any N.

3. Propose an implementation of ②.

4. Exemplify the potential of ②.
IDF and MED in information distance space

IDF of N-gram $g$ is the sum of IDF of $w_1, \ldots, w_N$ and MED of $g$

However, large MED means $g$’s collocation is unnatural.

$\text{IDF}(w_1, \ldots, w_N) = E(\mu(g), \phi(\varepsilon))$

$\text{MED}(g) = E(\phi(g), \mu(g))$

$\text{IDF}(g) = E(\phi(g), \phi(\varepsilon))$

$\mu(g)$: semantic of $g$

$\phi(g)$: context of $g$

$\phi(\varepsilon)$: context of $\varepsilon$
N-gram IDF

We redesign IDF for N-grams.
Larger IDF and smaller MED is better.

\[ IDF_{N-gram}(g) = IDF(w_1, \ldots, w_N) - MED(g) = \log \frac{|D| \cdot df(g)}{df(w_1, \ldots, w_N)^2} \]

[Bu+, COLING10]

\[ MED(g) = E(\phi(g), \mu(g)) \]

\[ IDF(w_1, \ldots, w_N) = E(\mu(g), \phi(\varepsilon)) \]

\[ IDF(g) = E(\phi(g), \phi(\varepsilon)) \]

\[ \phi(g): \text{context of } g \]

\[ \mu(g): \text{semantic of } g \]

\[ \phi(\varepsilon): \text{context of } \varepsilon \]
## Key Term Extraction using N-gram IDF

### Input: “Alice’s Adventures in Wonderland - Kindle edition by Lewis Carroll”

<table>
<thead>
<tr>
<th>N-gram</th>
<th>$IDF_{N-gram}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>kindle edition</td>
<td>12.043</td>
</tr>
<tr>
<td>kindle</td>
<td>11.653</td>
</tr>
<tr>
<td><strong>alice's adventures in wonderland</strong></td>
<td><strong>11.496</strong></td>
</tr>
<tr>
<td>adventures in wonderland</td>
<td>10.906</td>
</tr>
<tr>
<td>s adventures in wonderland</td>
<td>10.804</td>
</tr>
<tr>
<td>wonderland</td>
<td>9.670</td>
</tr>
<tr>
<td><strong>lewis carroll</strong></td>
<td><strong>9.498</strong></td>
</tr>
<tr>
<td>alice s adventures</td>
<td>9.385</td>
</tr>
<tr>
<td>alice s adventures in</td>
<td>9.348</td>
</tr>
<tr>
<td>in wonderland</td>
<td>8.762</td>
</tr>
<tr>
<td>carroll</td>
<td>8.152</td>
</tr>
<tr>
<td>by lewis carroll</td>
<td>7.461</td>
</tr>
<tr>
<td>alice</td>
<td>7.234</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N-gram</th>
<th>$IDF_{N-gram}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>adventures</td>
<td>7.101</td>
</tr>
<tr>
<td>kindle edition by</td>
<td>6.739</td>
</tr>
<tr>
<td>lewis</td>
<td>6.192</td>
</tr>
<tr>
<td>edition</td>
<td>4.836</td>
</tr>
<tr>
<td>adventures in</td>
<td>4.280</td>
</tr>
<tr>
<td>s adventures</td>
<td>3.586</td>
</tr>
<tr>
<td>alice s</td>
<td>3.507</td>
</tr>
<tr>
<td>s adventures in</td>
<td>2.255</td>
</tr>
<tr>
<td>by lewis</td>
<td>1.768</td>
</tr>
<tr>
<td>s</td>
<td>1.030</td>
</tr>
<tr>
<td>by</td>
<td>0.820</td>
</tr>
<tr>
<td>in</td>
<td>0.154</td>
</tr>
<tr>
<td>edition by</td>
<td>-0.875</td>
</tr>
</tbody>
</table>
## Contributions

<table>
<thead>
<tr>
<th>1</th>
<th>Give a new explanation of IDF.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IDF</strong> is equal to the distance between the term and the empty string in the information distance space.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>Propose a new IDF scheme that can handle N-grams of any N.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDF and MED are connected in the proposed scheme. Also, it is capable of extracting key N-grams from texts without using NLP techniques.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>Propose an implementation of ②.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>Exemplify the potential of ②.</th>
</tr>
</thead>
</table>
Calculation of N-gram IDF

Calculation of the document frequency of $w_1, \cdots, w_N$ (set of words) requires much computational cost. Bu et al. [COLING10] just used Web search engine for their experiments.

→ **Wavelet tree**
   [Gagie+, TCS, 12]

How to determine N is unclear.

→ **Suffix tree (or enhanced suffix array)**
   [Okanohara+, SDM09]

Please refer to our paper for the detail.
Implementation, Data, and Demo

Code to calculate N-gram IDF for all N-grams

https://github.com/iwnsew/ngweight

Processed English Wikipedia (Oct 1, 2013)

http://mljournalism.com/ngw/ngram.bz2

It took 12 days to process whole Wikipedia

Online demo

http://mljournalism.com/ngw/

Or, search “N-gram TF-IDF”.
Contributions

1. Give a new explanation of IDF.
   
   IDF is equal to the distance between the term and the empty string in the information distance space.

2. Propose a new IDF scheme that can handle N-grams of any N.
   
   IDF and MED are connected in the proposed scheme. Also, it is capable of extracting key N-grams from texts without using NLP techniques.

3. Propose an implementation of ②.
   
   Two cutting-edge string processing algorithms were combined. https://github.com/iwnsew/ngweight

4. Exemplify the potential of ②.
Evaluation 1: Key Term Extraction

Use 1,678 first paragraphs of English Wikipedia
Use Anchor texts and bold faces as correct labels R

<table>
<thead>
<tr>
<th>Method</th>
<th>R-Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>$TF-IDF_{N-gram}$</td>
<td>0.377</td>
</tr>
<tr>
<td>Noun Phrase + TF-IDF sum</td>
<td>0.386</td>
</tr>
<tr>
<td>[Hasan+, COLING10]</td>
<td></td>
</tr>
<tr>
<td>Noun Phrase (no capitalization) + TF-IDF sum</td>
<td>0.369</td>
</tr>
<tr>
<td>[Hasan+, COLING10]</td>
<td></td>
</tr>
<tr>
<td>Noun Phrase + TF-IDF average</td>
<td>0.367</td>
</tr>
<tr>
<td>Noun Phrase (no capitalization) + TF-IDF average</td>
<td>0.352</td>
</tr>
<tr>
<td>Noun Phrase + TF-IDF</td>
<td>0.369</td>
</tr>
<tr>
<td>Noun Phrase (no capitalization) + TF-IDF</td>
<td>0.355</td>
</tr>
<tr>
<td>TF-IDF (word only)</td>
<td>0.229</td>
</tr>
</tbody>
</table>

Just use the weight!

Require POS tags
Evaluation 2: Query Segmentation

Use IR-based query segmentation dataset [Roy+, SIGIR12]
Evaluate search results by using segmented phrases

<table>
<thead>
<tr>
<th>Method</th>
<th>nDCG (Top 5)</th>
<th>nDCG (Top 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(IDF_{N-gram})</td>
<td>0.730</td>
<td>0.742</td>
</tr>
<tr>
<td>Mishra (use query logs) [Mishra+, WWW11]</td>
<td>0.706</td>
<td>0.737</td>
</tr>
<tr>
<td>Mishra + Wikipedia titles [Roy+, SIGIR12]</td>
<td>0.725</td>
<td>0.750</td>
</tr>
<tr>
<td>PMI (use query logs) [Roy+, SIGIR12]</td>
<td>0.716</td>
<td>0.736</td>
</tr>
<tr>
<td>PMI (use Web corpus) [Roy+, SIGIR12]</td>
<td>0.670</td>
<td>0.707</td>
</tr>
<tr>
<td>No segmentation</td>
<td>0.655</td>
<td>0.689</td>
</tr>
</tbody>
</table>

Just use the weight! Require query logs or external knowledge
Contributions and Conclusion

1. Give a new explanation of IDF.
   
   IDF is equal to the distance between the term and the empty string in the information distance space.

2. Propose a new IDF scheme that can handle N-grams of any N.
   
   IDF and MED are connected in the proposed scheme. Also, it is capable of extracting key N-grams from texts without using NLP techniques.

3. Propose an implementation of 2.
   
   Two cutting-edge string processing algorithms were combined.

   https://github.com/iwnsew/ngweight

4. Exemplify the potential of 2.
   
   On key term extraction and query segmentation tasks, N-gram IDF achieved competitive performance with task-oriented methods.
Future work

Efficient computation of N-gram IDF

Supporting languages without spaces between words such as Japanese and Chinese

Theoretical explanation of TF
<table>
<thead>
<tr>
<th>1</th>
<th>Give a new explanation of IDF.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IDF is equal to the distance between the term and the empty string in the information distance space.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>Propose a new IDF scheme that can handle N-grams of any N.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IDF and MED are connected in the proposed scheme. Also, it is capable of extracting key N-grams from texts without using NLP techniques.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>Propose an implementation of 2.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Two cutting-edge string processing algorithms were combined. <a href="https://github.com/iwnsew/ngweight">https://github.com/iwnsew/ngweight</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>Exemplify the potential of 2.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>On key term extraction and query segmentation tasks, N-gram IDF achieved competitive performance with task-oriented methods.</td>
</tr>
</tbody>
</table>
Suffix tree for enumerating valid N-grams
[Okanohara+, SDM09]

Intermediate nodes having multiple prefixes = valid N-gram
Number of valid N-grams is proved to be linearly proportional to text length.

Text: “to be or not to be to live or to die”

Position: 1 5 10 7 3 2 8 0 4 9 6
Prefix: to to to to or be live not or be
Wavelet tree for DF counting

[Gagie+, TCS, 12]

The most efficient algorithm for counting DF for a set of words

Document set: \( D = \{a, b, c, d\} \)

\( a = \text{“to be”}, \ b = \text{“or not to be”}, \ c = \text{“to live”}, \ d = \text{“or to die”} \)

Position: 1 5 10 7 3 2 8 0 4 9 6

Document ID: a b d c b b d a b d c

\[
\begin{align*}
\text{abdcbbdabdc} \\
00110010011 \\
\begin{array}{l}
\text{abbbab} \\
011101 \\
\text{dcdddc} \\
10110 \\
\end{array}
\end{align*}
\]

\( \approx \log|D| \)
Example of DF counting

Keep beginning and end positions of each word, and traverse the tree toward leaves.

**Document set:** $D = \{a, b, c, d\}$  
\[ a = \text{“to be”}, \quad b = \text{“or not to be”}, \quad c = \text{“to live”}, \quad d = \text{“or to die”} \]

**Query:** “to” “be”  
**Results:** a, b