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

Masumi Shirakawa
Osaka University
1-5 Yamadaoka, Suita
Osaka 565-0871, Japan
shirakawa.masumi@ist.osaka-u.ac.jp

Takahiro Hara
Osaka University
1-5 Yamadaoka, Suita
Osaka 565-0871, Japan
hara@ist.osaka-u.ac.jp

Shojiro Nishio
Osaka University
1-5 Yamadaoka, Suita
Osaka 565-0871, Japan
nishio@ist.osaka-u.ac.jp

## ABSTRACT

This paper first reveals the relationship between Inverse Document Frequency (IDF), a global term weighting scheme, and information distance, a universal metric defined by Kolmogorov complexity. We concretely give a theoretical explanation that the IDF of a term is equal to the distance between the term and the empty string in the space of information distance in which the Kolmogorov complexity is approximated using Web documents and the Shannon-Fano coding. Based on our findings, we propose *N-gram IDF*, a theoretical extension of IDF for handling words and phrases of any length. By comparing weights among N-grams of any $N$, N-gram IDF enables us to determine dominant N-grams among overlapping ones and extract key terms of any length from texts without using any NLP techniques. To efficiently compute the weight for all possible N-grams, we adopt two string processing techniques, i.e., maximal substring extraction using enhanced suffix array and document listing using wavelet tree. We conducted experiments on key term extraction and Web search query segmentation, and found that N-gram IDF was competitive with state-of-the-art methods that were designed for each application using additional resources and efforts. The results exemplified the potential of N-gram IDF.

## Categories and Subject Descriptors

H.1.1 [**Models and Principles**]: Systems and Information Theory—*information theory*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

## General Terms

Experimentation, Measurement, Theory

## Keywords

Term Weighting; IDF; Multiword Expression; MED; Information Distance; Kolmogorov Complexity

*WWW 2015,* May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3469-3/15/05.
http://dx.doi.org/10.1145/2736277.2741628.

## 1. INTRODUCTION

Term weighting schemes as represented by TF-IDF [42], short for Term Frequency-Inverse Document Frequency, are fundamental technologies for text analysis. TF-IDF was originally introduced as a weighting factor of each word in document retrieval where a document is represented by a vector of words that occur in it. Afterwards TF-IDF has become a de facto standard of term weighting scheme for bag-of-words representation of text documents in information retrieval and text mining. Today it has been also used for explicitly highlighting key terms in texts in the area of natural language processing [19].

Term weighting schemes can be usually decomposed by two components: local term weighting scheme and global term weighting scheme. Local term weighting schemes give a weight to a term using its local document information such as term frequency and term co-occurrence in a target document. The local weight of a certain term varies depending on the document where the term occurs. Global term weighting schemes use global document information such as document frequency and total term frequency over documents. The global weight of a certain term is fixed, that is, it is independent from the target document in which the term occurs.

Representative term weighting schemes include TF-IDF, Okapi BM25 [37] and recently proposed TW-IDF [39]. Each weight for term $t$ in document $d \in D$ is specifically computed as below

$$TF\text{-}IDF(t,d) = tf(t,d) \cdot \log \frac{|D|}{df(t)}$$

$$BM25(t,d) = \frac{(k_1+1) \cdot tf(t,d)}{k_1 \cdot (1-b+b \cdot \frac{|d|}{avdl}) + tf(t,d)} \cdot \log \frac{|D|}{df(t)}$$

$$TW\text{-}IDF(t,d) = \frac{tw(t,d)}{1-b+b \cdot \frac{|d|}{avdl}} \cdot \log \frac{|D|}{df(t)}$$

where $tf(t,d)$ is the term frequency of $t$ in $d$, $df(t)$ is the document frequency of $t$ over document set $D$, $|D|$ is the cardinality of $D$ (i.e., total number of documents in $D$), $|d|$ is the length of $d$ (i.e., total number of words in $d$), $avdl$ is the average length of documents in $D$, $tw(t,d)$ is a graph-based $tf$-like function, and $k_1$ and $b$ are constant values[1]. Whereas they are very different regarding their local term

[1] Reasonable values are to set $k_1$ to a value between 1.2 and 2, and $b = 0.75$ [26].

weighting schemes, they use the same global term weighting scheme, i.e., IDF [21].

$$IDF(t) = \log \frac{|D|}{df(t)} \tag{1}$$

The reason why IDF is adopted in the representative term weighting schemes lies in its simplicity and robustness. The simplicity is easily understandable from Eq. (1). Even as the simplicity, IDF has proven to be justified and robust by the literature [2, 16, 27, 32, 36]. There are some choices of global term weighting schemes such as Residual IDF (RIDF) [8] and gain [32] other than IDF. IDF, however, has been chosen in many cases because of its simplicity (i.e., one can easily use it without the knowledge) and robustness (i.e., it works reasonably well for most applications).

One of the main drawbacks of IDF is that it cannot handle N-grams for $N > 1$, or phrases, which are composed of two or more words. IDF gives more weight to terms occurring in less documents. However, phrases occur in less documents when their collocations are more awkward. Consequently, awkward phrases unintentionally gain much weight. For example, estimated document frequencies of "Osaka University" and "Osaka be" using Google Search[2] were 1,890,000 and 6,160 respectively, resulting in that the latter gained far more IDF weight. The definition of IDF thus totally acts counter to the definition of good phrases.

Aside from term weighting schemes, a considerable number of studies have been made on extracting phrases, or Multiword Expressions (MWEs). Pointwise Mutual Information (PMI) [9] and Multiword Expression Distance (MED) [7] are representative solid schemes to measure a score that defines how likely a sequence of consecutive words is to compose a phrase. The score just indicates the compositionality of words, that is, it does not indicate how important the composed phrase is. To the best of our knowledge, there is no theoretical explanation that deals with both term weighting and multiword expression extraction. Weighting terms of any length so far requires heuristics.

This paper tackles a challenge of bridging the theoretical gap between term weighting and multiword expression extraction. In particular, we connect IDF and MED in the space of information distance [3]. Information distance is a universal metric defined by Kolmogorov complexity [22, 25]. MED is a theoretically justified information distance-based metric for multiword expression extraction that works better than PMI and several heuristic measures. We evince the relationship between IDF and information distance to connect it with MED. Our findings enable us to design schemes for handling both problems, i.e., term weighting and multiword expression extraction at the same time.

We also propose a term weighting scheme, *N-gram IDF*, that can handle terms of any length by combining IDF and MED in the space of information distance. N-gram IDF is capable of weighting phrases (i.e., N-grams for $N > 1$) as well as words in a sole theoretical scheme. It is therefore able to compare weights among words and phrases. Using N-gram IDF, we can obtain dominant N-grams among overlapping ones and extract key terms of any length from texts without using NLP techniques. We verify the simplicity and robustness of N-gram IDF by two applications: key term extraction and Web search query segmentation.

The remainder of this paper is organized as follows: Section 2 describes related work on term weighting and multiword expression extraction. Section 3 gives a yet another justification of IDF using Kolmogorov complexity and information distance. Based on our findings presented in Section 3, we propose N-gram IDF in Section 4. Section 5 illustrates how to compute N-gram IDF using efficient data structures that have been developed in the area of string processing. We show the potential of N-gram IDF on a couple of applications in Section 6. Finally, we conclude our work in Section 7.

## 2. RELATED WORK

Term weighting schemes are vital for text-related applications and hence have been well studied. TF-IDF [41] is the most popular (general-purpose) term weighting scheme among representative ones, followed by Okapi BM25 [37]. In addition to the effectiveness of TF-IDF in various applications such as information retrieval [46], document clustering [14], key term extraction [19] and object matching in videos [44], many theoretical explanations [2, 20, 36, 38] encourage researchers and developers to employ TF-IDF. Based on or inspired by TF-IDF, heuristically improved schemes such as TW-IDF [39] have been proposed.

The most popular global term weighting scheme is IDF [21], which has been adopted by TF-IDF, BM25 and TW-IDF. There are many alternatives such as $x^I$ [5], z-measure [18], Residual IDF (RIDF) [8] and gain [32]. However, IDF is still a de facto standard of global term weighting scheme. This is due to the simplicity and robustness of IDF against the alternatives. Even as its simplicity of just computing $\log \frac{|D|}{df(t)}$, IDF has many justifications [2, 16, 27, 32, 36] to back up its robustness. In other words, it is difficult to beat IDF without using heuristics. Some reported that RIDF was superior to IDF in certain applications [31, 35]. This supports that RIDF can be a good alternative of IDF, though it is heuristic and the performance in other applications or datasets is not guaranteed. Specialized global term weighting schemes like Inverse Corpus Frequency (ICF) [34], designed for analyzing text stream, and Relevant Frequency (RF) [23], for text classification, can also be alternatives in the target applications.

As can be seen from the fact that there are many alternatives, IDF has some drawbacks to be improved. Among them, critical, but not solved one is that it is unable to handle phrases. Handling phrases requires the measurement of word compositionality, i.e., measuring how much the collocation is natural, as well as the term weighting. The measurement of the word compositionality, or Multiword Expression (MWE) extraction, has been an object of study for a long time. Pointwise Mutual Information (PMI) [9] and their variations are popular and effective for measuring the word compositionality [6, 12, 13, 43, 48]. Theoretically defined PMI was originally developed for measuring the association between a couple of words. Among 84 measures, PMI was the best for measuring the bi-gram compositionality [33]. When it comes to N-grams for $N > 2$, PMI needs to be extended to cope with them. Some literature heuristically extends PMI by computing the arithmetic average of [13, 12] or the best score among every possible separation [43]. Enhanced Mutual Information (EMI) [48] is another extension of PMI, measuring the cohesion of an N-gram by using

[2] Results were retrieved on November 9, 2014.

the frequency of each word. Symmetric Conditional Probability (SCP) [12] is a measure similar to PMI and extended for N-grams for $N > 2$ by using the arithmetic average.

Recently proposed Multiword Expression Distance (MED) [7] is a theoretically justified measure of the word (non-)compositionality for N-grams for $N > 1$ based on information distance [3]. It has proven to be better than the heuristic extensions of PMI and SCP described above. Since MED is deeply related to our work, we minutely describe it in Section 3. At this time, we can conclude that MED is the most promising measure of the word compositionality based on a solid theory.

Term weighting and multiword expression extraction, as above, have been separately studied. One is for weighting terms and the other is for measuring the compositionality of consecutive words. No theoretical explanation, however, has been given to handle both problems. In this work, we first give a theoretical explanation that connects IDF, a global term weighting scheme, and MED, a measure for multiword expression extraction.

## 3. IDF AND INFORMATION DISTANCE

In this section, we uncover the relationship between IDF and information distance through explaining Kolmogorov complexity, information distance and Multiword Expression Distance (MED).

### 3.1 Kolmogorov Complexity

Kolmogorov complexity [22, 25] is a measure of the randomness of a (bit) string. It is also known as descriptive complexity, algorithmic entropy or program-size complexity, namely indicating the minimum amount of resources to describe a string on a universal Turing machine. We define $K(x)$ as the Kolmogorov complexity of string $x$. $K(x)$ is specifically the length of the shortest program that outputs $x$. For example, given strings $x_1$ and $x_2$,

$$\begin{aligned} x_1 &= \text{“0101010101010101 00”} \\ x_2 &= \text{“011101100010110010”} \end{aligned}$$

$x_1$ can be shortly described as “01” ×8+ “00” whereas $x_2$ seems to be difficult to shorten. We can thus estimate that $K(x_1)$ is smaller than $K(x_2)$. Because the exact value of the Kolmogorov complexity is not computable, it is usually approximated by using compression algorithms [10] or Web documents [11].

We also define $K(x|y)$ as the conditional Kolmogorov complexity of string $x$ given another string $y$. $K(x|y)$ is the length of the shortest program that outputs $x$ from input $y$. Given that $\epsilon$ is the empty string, $K(x)$ can be written as $K(x|\epsilon)$. Given strings $x_1$, $x_2$ and $y$,

$$\begin{aligned} x_1 &= \text{“010101010101010100”} \\ x_2 &= \text{“011101100010110010”} \\ y &= \text{“01110110001011001”} \end{aligned}$$

$x_2$ can be efficiently described as $y+$ “0” using $y$. Consequently, $K(x_2|y)$ might be slightly smaller than $K(x_1|y)$. It is noteworthy that $K(x, y)$, the Kolmogorov complexity of the concatenated string of strings $x$ and $y$, is expressed as

$$K(x, y) = K(x|y) + K(y) = K(y|x) + K(x) \tag{2}$$

up to an additive logarithmic term. Eq. (2) intuitively means that one string can be reused to describe the other.

### 3.2 Information Distance

Information distance [3] is a universal metric defined by the Kolmogorov complexity. It is an application-independent, unique objective distance just exactly like the distance in the physical world. It is actually the energy cost for transforming one string to the other. According to Landauer's principle [24], irreversibly processing one bit of information costs $1k\mathcal{T} \cdot ln(2)$ where $k$ is the Boltzmann constant and $\mathcal{T}$ is the absolute temperature in Kelvin. Based on the Landauer's principle and the Kolmogorov complexity, information distance $E(x, y)$ between two strings $x$ and $y$ is defined as

$$E(x, y) = \max\{K(x|y), K(y|x)\} \tag{3}$$

up to an additive logarithmic term. Eq. (3) is transformed into the following equation using Eq. (2).

$$E(x, y) = K(x, y) - \min\{K(y), K(x)\} \tag{4}$$

Information distance has proven to be a metric, or a distance function, i.e., it satisfies non-negativity, identity of indiscernibles, symmetry and triangle inequality, which are respectively represented by the following formulas.

$$\begin{aligned} &E(x, y) \geq 0 \\ &E(x, y) = 0 \Leftrightarrow x = y \\ &E(x, y) = E(y, x) \\ &E(x, z) \leq E(x, y) + E(y, z) \end{aligned}$$

Moreover, information distance has been shown to be universal, or optimal. Distance $\mathcal{D}(x, y)$ is said to be admissible (i.e., an upper-semicomputable and normalized metric) if it satisfies the following density condition (Kraft's inequality).

$$\sum_{y:y\neq x} 2^{-\mathcal{D}(x,y)} \leq 1, \quad \sum_{x:x\neq y} 2^{-\mathcal{D}(x,y)} \leq 1$$

The density condition restricts the number of objects within a given distance from an object. When $\mathcal{D}(x, y)$ is admissible, there is a constant $C$ for all $x$ and $y$, and

$$E(x, y) \leq \mathcal{D}(x, y) + C.$$

Thus, $E(x, y)$ minorizes every admissible distance $\mathcal{D}(x, y)$ up to an additive constant, indicating that information distance is universal.

Information distance is generally utilized for measuring the similarity between two objects. Because the exact value of the information distance is not computable as well as the Kolmogorov complexity, it is computed using approximations such as Normalized Compression Distance (NCD) [10] and Normalized Google Distance (NGD) [11]. NCD measures the compression size of the concatenated string of $x$ and $y$ versus that of each string. NGD is intended for texts, counting the number of Web pages containing both terms $x$ and $y$ versus the number of Web pages containing each term. Cilibrasi and Vitányi [11] reported that the distance estimated by NGD was stable for the growing Web. It can be guessed that the stability comes from the universality of information distance.

### 3.3 Multiword Expression Distance

Multiword Expression Distance (MED) [7] is a universal metric for measuring the word compositionality based on the information distance. It in particular computes the information distance between the context and semantic of an N-gram. Inspired by Cilibrasi and Vitányi [11], Bu et al. [7] defined the context of an N-gram as the set of Web pages containing it and the semantic of an N-gram as the set of Web pages containing every word composing it. Obviously, the semantic of an N-gram subsumes the context of the N-gram. For example, the semantic of “football player” includes not only Web pages containing itself but also those containing “football” and “player.”

Let us formulate MED according to Bu et al. [7]. We denote $w$ as a word (uni-gram), $W$ as a set of all words, $g$ as an N-gram, $G \equiv W^+$ as a set of all N-grams, $D$ as a set of all Web pages (documents), $t$ as a search term that is an N-gram or the conjunction of search terms and $T$ as a set of all search terms (namely, $G \subset T$). Let $\phi : T \to 2^D$ be the context function that maps a search term $t$ to a set of Web pages containing all the N-grams in $t$, denoted by $\phi(t)$. Let $\theta : G \to T$ be the function that maps an N-gram $g = w_1 ... w_N$ to $\bigwedge_{i=1}^{N} w_i$, the conjunction of words composing $g$, denoted by $\theta(g)$. Let $\mu : G \to 2^D$ be the semantic function that is a composite function $\phi \circ \theta$, that is, $\mu(g) = \phi(\theta(g))$. From the definitions, we have $\phi(g) \subseteq \mu(g)$. Given an N-gram $g$, MED is defined as the information distance between $\phi(g)$, the context of $g$, and $\mu(g)$, the semantic of $g$. Using Eq. (4), MED is formulated as follows.

$$\begin{aligned} MED(g) &= E(\phi(g), \mu(g)) \\ &= K(\phi(g), \mu(g)) - \min\{K(\phi(g)), K(\mu(g))\} \quad (5) \end{aligned}$$

To approximately compute the Kolmogorov complexity $K(x)$, MED utilizes Shannon-Fano coding to encode the probability of $x$. We assume that all Web pages are equiprobable, i.e., the probability that a Web page is chosen is $\frac{1}{|D|}$. Let $p : \phi(T) \to [0, 1]$ be the context probability function where $\phi(T)$ is a set of all contexts, namely $\phi(T) \equiv \{x | \exists y \in T, x = \phi(y)\}$. Because each context is a set of Web pages, the probability of context $c$ is defined as

$$p(c) = \frac{|c|}{M} \quad (6)$$

where $M = \sum_{c_i \in \phi(T)} p(c_i)$. This informally says that the probability that a set of Web pages $c$ is chosen is proportional to the cardinality of $c$. Consequently, the Kolmogorov complexity $K$ can be approximated by using the Shannon-Fano code [25] length associated with $p$.

$$K(x) \approx -\log p(x) \quad (7)$$

$$K(x, y) \approx -\log p(x, y) \quad (8)$$

Using Eqs. (6), (7) and (8), Eq. (5) is approximated as

$$\begin{aligned} &MED(g) \\ &\quad \approx \max\{\log p(\phi(g)), \log p(\mu(g))\} - \log p(\phi(g), \mu(g)) \\ &\quad = \max\{\log |\phi(g)|, \log |\mu(g)|\} - \log M \\ &\qquad - \log |\phi(g) \cap \mu(g)| + \log M \\ &\quad = \max\{\log |\phi(g)|, \log |\mu(g)|\} - \log |\phi(g) \cap \mu(g)|. \quad (9) \end{aligned}$$

Since $\phi(g) \subseteq \mu(g)$, Eq. (9) is finally

$$MED(g) \approx \log |\mu(g)| - \log |\phi(g)| = \log \frac{|\mu(g)|}{|\phi(g)|}. \quad (10)$$

Eq. (10) can be computed using the cardinality of $\phi(g)$ and $\mu(g)$. Specifically, $|\phi(g)|$ is the document frequency of $g$ and $|\mu(g)|$ is the document frequency of $\theta(g)$, a conjunction of words composing $g$. In the literature [7], $|\phi(g)|$ and $|\mu(g)|$ are estimated by using a general search engine. The document frequency of $\theta(g)$ is obtained from the query of “logic and” of each word $w_i$ in $g$.

## 3.4 Inverse Document Frequency

Here, we reveal the relationship between IDF and information distance based on the above discussions. Just like Eq. (1), we define IDF of an N-gram $g$ as follows.

$$IDF(g) = \log \frac{|D|}{df(g)} = \log \frac{|D|}{|\phi(g)|} \quad (11)$$

We can be aware that IDF (Eq. (11)) and MED (Eq. (10)) are analogous to each other in their mathematical forms. The difference is that the numerator in the logarithmic term is $|D|$ in IDF contrast to $|\mu(g)|$ in MED. Are there any relationships between IDF and information distance? We further investigate the answer to this question.

Let us denote the empty string, or zero-gram, by $\epsilon$. Define $\epsilon \in G$ and $G \equiv W^*$. $\phi(\epsilon)$ is a set of Web pages containing $\epsilon$. Clearly, $\epsilon$ is contained in all Web pages. $\phi(\epsilon)$ is therefore equal to $D$, a set of all Web pages. We then derive the information distance between N-gram $g$ and the empty string $\epsilon$. $E(\phi(g), \phi(\epsilon))$ is specifically defined as

$$E(\phi(g), \phi(\epsilon)) = K(\phi(g), \phi(\epsilon)) - \min\{K(\phi(g)), K(\phi(\epsilon))\}. \quad (12)$$

As well as the derivation of MED, Eq. (12) can also be transformed into

$$\begin{aligned} &E(\phi(g), \phi(\epsilon)) \\ &\quad \approx \max\{\log p(\phi(g)), \log p(\phi(\epsilon))\} - \log p(\phi(g), \phi(\epsilon)) \\ &\quad = \max\{\log |\phi(g)|, \log |\phi(\epsilon)|\} - \log M \\ &\qquad - \log |\phi(g) \cap \phi(\epsilon)| + \log M \\ &\quad = \max\{\log |\phi(g)|, \log |\phi(\epsilon)|\} - \log |\phi(g) \cap \phi(\epsilon)|. \quad (13) \end{aligned}$$

Because apparently $\phi(g) \subseteq \phi(\epsilon)$, Eq. (13) becomes

$$\begin{aligned} E(\phi(g), \phi(\epsilon)) &\approx \log |\phi(\epsilon)| - \log |\phi(g)| \\ &= \log \frac{|\phi(\epsilon)|}{|\phi(g)|} \\ &= \log \frac{|D|}{|\phi(g)|} \\ &= IDF(g). \end{aligned}$$

We finally obtain the relationship between IDF and information distance, i.e., IDF of an N-gram $g$ is equal to the information distance between $\phi(g)$ and $\phi(\epsilon)$.

Let us summarize our findings on IDF and information distance. In the space of information distance in which the Kolmogorov complexity is approximated using Web documents and the Shannon-Fano coding, the IDF of a term is equal to the distance between the term and the empty string, which is normally the base of the Kolmogorov complexity. The information distance can therefore be used as the global weight of terms. The farther a term is from the empty string in the space of information distance, the larger its weight is. We can henceforth design term weighting schemes in the space of information distance. Our findings are substantially useful because it becomes far easier to combine term weighting with MED, an information distance-based metric for multiword expression extraction. In next section, we show a theoretically justified combination of IDF and MED for weighting N-grams of any $N$.

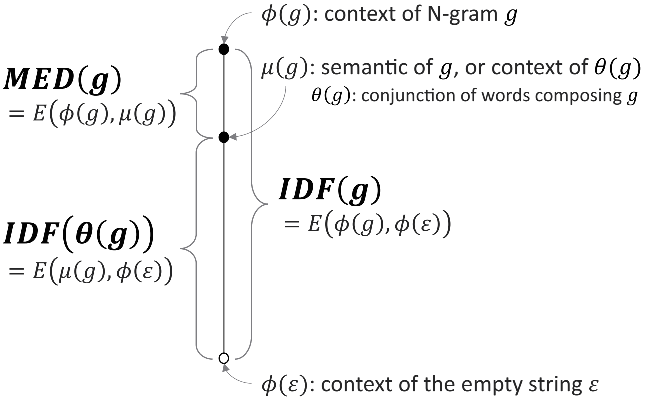
$\phi(g)$: context of N-gram $g$
$\mu(g)$: semantic of $g$, or context of $\theta(g)$
$\theta(g)$: conjunction of words composing $g$
$\boldsymbol{MED(g)}$
$= E(\phi(g), \mu(g))$
$\boldsymbol{IDF(g)}$
$= E(\phi(g), \phi(\varepsilon))$
$\boldsymbol{IDF(\theta(g))}$
$= E(\mu(g), \phi(\varepsilon))$
$\phi(\varepsilon)$: context of the empty string $\varepsilon$

**Figure 1: Relationship between IDF and MED in the space of information distance where the Kolmogorov complexity is approximated using Web documents and the Shannon-Fano coding. $\phi(g)$, $\mu(g)$ and $\phi(\epsilon)$ are on a line, i.e., the equity holds in the triangle inequity.**

## 4. N-GRAM IDF

We propose a global term weighting scheme that is capable of handling terms of any length, namely N-grams for $N \geq 1$, based on the information distance. In Section 3, we revealed that the IDF of an N-gram is equal to the information distance between the N-gram and the empty string when the Kolmogorov complexity is approximated using Web documents and the Shannon-Fano coding. Similarly, the MED of an N-gram is the information distance between the N-gram and the conjunction of words composing the N-gram. Figure 1 shows the relationship between IDF and MED in the space of information distance. In this space, N-gram $g$, the conjunction of words $\theta(g)$ and the empty string $\epsilon$ are represented as the sets of Web pages $\phi(g)$, $\mu(g)$ and $\phi(\epsilon)$ respectively. Note that the equity holds in the triangle inequity, i.e., $E(\phi(g), \phi(\epsilon)) = E(\phi(g), \mu(g)) + E(\mu(g), \phi(\epsilon))$.

$$
\begin{aligned}
E(\phi(g), \mu(g)) + E(\mu(g), \phi(\epsilon)) &= \log \frac{|\mu(g)|}{|\phi(g)|} + \log \frac{|D|}{|\mu(g)|} \\
&= \log \frac{|D|}{|\phi(g)|} \\
&= E(\phi(g), \phi(\epsilon))
\end{aligned}
$$

The challenge in this work is to theoretically connect the term weighting and multiword expression extraction, both of which have been separately studied for a long time. IDF gives too much weight to an N-gram for $N > 1$ when the collocation of the N-gram is awkward. In that case, MED should also be large since it exactly measures how much the collocation of the N-gram is awkward. Taken together, it seems reasonable to give more weight to an N-gram when its IDF is large and its MED is small. We can consider two of such schemes in the space of information distance (Figure 1): one is using $IDF(\theta(g))$ instead of $IDF(g)$ and the other is using both $IDF(\theta(g))$ and $MED(g)$.

$IDF(\theta(g))$ is the IDF of $\theta(g)$, the conjunction of words composing N-gram $g$. As can be seen in Figure 1, it is equivalent to $IDF(g) - MED(g)$. In other words, it is the information distance between $\theta(g)$ and the empty string $\epsilon$. In particular, $IDF(\theta(g))$ is computed as

$$IDF(\theta(g)) = \log \frac{|D|}{|\mu(g)|} = \log \frac{|D|}{df(\theta(g))}. \tag{14}$$

This can give somewhat reasonable weight to N-grams. However, the weight monotonically grows as $N$ increases, that is, the weight for N-grams of different $N$ cannot be compared.

The other, and more promising, scheme is $IDF(\theta(g)) - MED(g)$. We can explain this by "how much the distance from the empty string to the semantic of $g$ can be shortened by giving the context of $g$." In Section 3, we revealed that the distance in the space of information distance represents the weight of a term. Here, we measure the distance in an indirect manner. We name it N-gram IDF, and it is specifically computed as follows.

$$
\begin{aligned}
IDF_{N\text{-}gram}(g) &= IDF(\theta(g)) - MED(g) \\
&= \log \frac{|D|}{|\mu(g)|} - \log \frac{|\mu(g)|}{|\phi(g)|} \\
&= \log \frac{|D|}{|\mu(g)|} + \log \frac{|\phi(g)|}{|\mu(g)|} \\
&= \log \frac{|D| \cdot |\phi(g)|}{|\mu(g)|^2} \\
&= \log \frac{|D| \cdot df(g)}{df(\theta(g))^2}
\end{aligned} \tag{15}
$$

When $N = 1$, both $IDF_{N\text{-}gram}(g)$ and $IDF(\theta(g))$ correspond to $IDF(g)$ because $\theta(g) = g$, i.e., $df(\theta(g)) = df(g)$.

We found that the weight given by N-gram IDF (Eq. (15)) is surprisingly stable for any $N$. Table 1 shows examples of N-gram IDF for a couple of texts (the weight is computed in Section 5). An important feature of N-gram IDF is the comparability of weights among N-grams of different $N$. We can therefore establish the dominance relationship among weights of all N-grams in a text to handle overlapping N-grams. We define dominant N-grams as those having the maximum weight of at least one position (word) of a given text, except those having the maximum weight of a single stop word, as shown in Figure 2. It rarely occurs that overlapping N-grams have the same weight. We can introduce a simple rule that adopts longer and prefix match among them. The number of dominant N-grams does not exceed the length of the text because one position of the text corresponds to a dominant N-gram. From Table 1, we can confirm that N-gram IDF weights are stable regardless of $N$ and dominant N-grams are well determined across different $N$.

## 5. IMPLEMENTATION

The computation of MED or N-gram IDF is not easy at all. Specifically, computing the document frequency of "logic and" of words requires much computational cost. Bu et al. [7] adopted an ad hoc approach that utilizes a general Web search engine to estimate the document frequency of the conjunction of words. However, it is difficult to compute them for all possible N-grams. Also, how to know all possible N-grams in a text corpus is an issue.

In order to solve the problems, we introduce two string processing techniques. First, we use enhanced suffix array [1] (as an alternative of suffix tree) to enumerate valid N-grams of any length. The idea originates from the equivalence class

**Table 1: Examples of N-gram IDF. Dominant N-grams are indicated in boldface with asterisk. Detail of the dominant N-grams for the left text is represented in Figure 2.**

| Alice's Adventures in Wonderland - Kindle edition by Lewis Carroll | | Fossil fuels must be phased out "almost entirely" by 2100 to avoid dangerous climate change | |
|---|---|---|---|
| ***kindle edition** | 12.043 | ***fossil fuels** | 11.211 |
| kindle | 11.653 | 2100 | 10.772 |
| ***alice s adventures in wonderland** | 11.496 | fuels | 9.752 |
| adventures in wonderland | 10.906 | ***phased** | 9.391 |
| s adventures in wonderland | 10.804 | phased out | 9.291 |
| wonderland | 9.670 | fossil | 8.332 |
| ***lewis carroll** | 9.498 | ***dangerous climate change** | 8.249 |
| alice s adventures | 9.385 | climate change | 7.432 |
| alice s adventures in | 9.348 | be phased out | 6.828 |
| in wonderland | 8.762 | by 2100 | 6.814 |
| carroll | 8.152 | be phased | 6.783 |
| by lewis carroll | 7.461 | dangerous | 6.749 |
| alice | 7.234 | climate | 6.575 |
| adventures | 7.101 | dangerous climate | 6.549 |
| kindle edition by | 6.739 | ***almost entirely** | 6.118 |
| lewis | 6.192 | ***avoid** | 6.063 |
| edition | 4.836 | entirely | 5.973 |
| adventures in | 4.280 | to avoid | 5.831 |
| s adventures | 3.586 | 2100 to | 5.031 |
| alice s | 3.507 | ***must** | 4.703 |
| s adventures in | 2.255 | change | 4.646 |
| by lewis | 1.768 | almost | 4.469 |
| s | 1.030 | fuels must | 4.283 |
| by | 0.820 | must be phased | 4.013 |
| in | 0.154 | must be phased out | 4.010 |
| edition by | -0.875 | must be | 3.831 |
| | | fuels must be | 3.478 |
| | | avoid dangerous | 2.998 |
| | | out | 2.612 |
| | | to avoid dangerous | 2.575 |
| | | entirely by | 2.055 |
| | | almost entirely by | 1.984 |
| | | be | 1.860 |
| | | by | 0.820 |
| | | to | 0.505 |
| | | out almost | -2.826 |

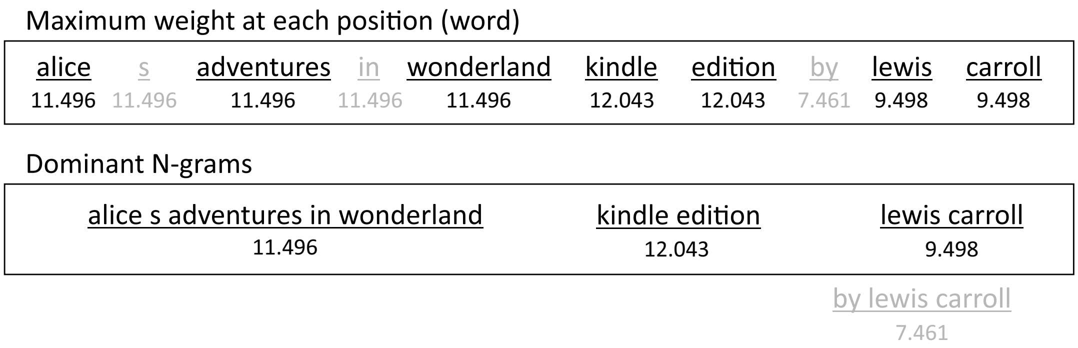

**Figure 2: Example of dominant N-grams for the left text of Table 1. "by lewis carroll" is not a dominant N-gram because it only dominates single stop word "by."**

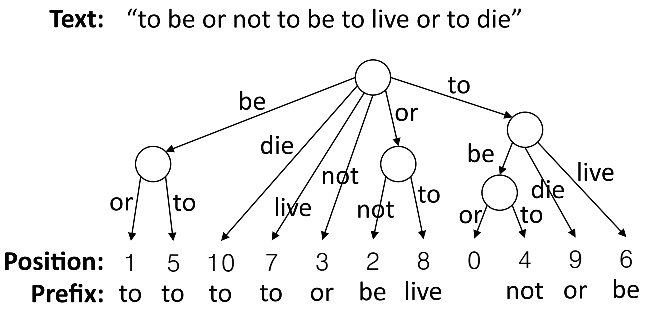
Text: "to be or not to be to live or to die"
be
to
die
or
not
live
not
to
be
die
live
or
to
or
to
Position: 1 5 10 7 3 2 8 0 4 9 6
Prefix: to to to to or be live not or be

**Figure 3: Example of enhanced suffix array (as an alternative of suffix tree). Intermediate nodes having multiple child nodes are candidates of the maximal substring ("be," "or," "to" and "to be"). Because maximal substring should also have multiple prefixes, "be" is not a maximal substring.**

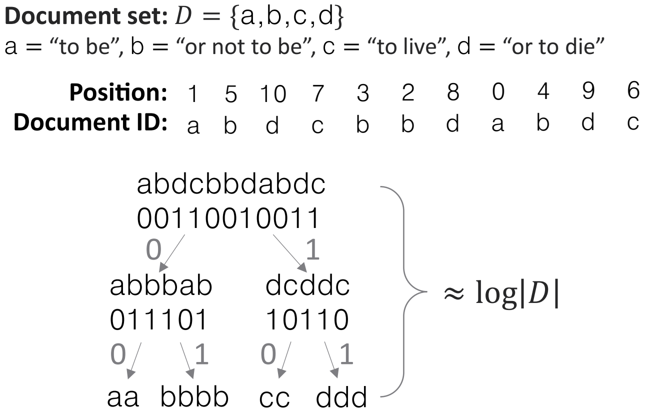
Document set: $D = \{a,b,c,d\}$
a = "to be", b = "or not to be", c = "to live", d = "or to die"
Position: 1 5 10 7 3 2 8 0 4 9 6
Document ID: a b d c b b d a b d c
abdcbbdabdc
00110010011
0
1
abbbab
011101
dcddc
10110
0
1
0
1
aa bbbb cc ddd
≈ log|D|

**Figure 4: Example of wavelet tree for document IDs.**

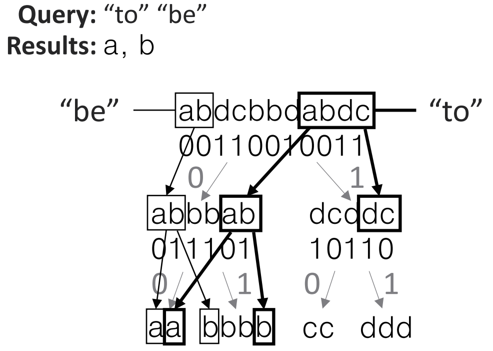

**Figure 5: Example of wavelet tree-based document listing for the conjunction of words as a query.**

of substrings that occur in (roughly) the same positions and have the same frequency [4]. It is guaranteed that the number of equivalence classes is less than twice of the length of a text. The literature [29, 30] proposed efficient algorithms to enumerate the longest substring (maximal substring) for every equivalence class in linear time using suffix tree or enhanced suffix array. Maximal substrings should be intermediate nodes having multiple child nodes in the suffix tree. Figure 3 illustrates an example of the suffix tree for text "to be or not to be to live or to die." Among all N-grams, "be," "or," "to" and "to be" have multiple child nodes. Maximal substrings should also have multiple prefixes. By memorizing prefixes for every N-gram, we can find that the prefix of "be" is unique, i.e., "to." Consequently, we obtain three maximal substrings "or," "to" and "to be." In this work, we adopt the algorithm [30] and library esaxx[3] to obtain all maximal substrings as valid N-grams.

Second, we utilize wavelet tree [17] for counting the document frequency of the conjunction of words. Wavelet tree is a succinct data structure that has recently been used for various purposes. Gagie et al. [15] showed that it can be used for document listing or counting the document frequency. While suffix array can be used for counting the document frequency of all N-grams [47], wavelet tree also enables us to count the document frequency of the conjunction of words. The time complexities are $O(df(g) \cdot \log|D|)$ for N-gram $g$ and $O(N \cdot df(\theta(g)) \cdot \log|D|)$ for $\theta(g)$, the conjunction of words composing $g$. According to the literature [15], the latter time complexity is close to the lower bound. We adopt the document listing algorithm [15] and use library wat-array[4].

Let us describe the process flow to compute the N-gram IDF weight for all valid N-grams. Given a set of documents $D$ as a single concatenated text[5], our implementation first enumerates valid N-grams by building the enhanced suffix array for the input text. Along with the first process, it obtains a list of document identifiers (document IDs) sorted by texts, not by document IDs. By doing this, we can represent all documents containing a certain N-gram by a region in the list. It next builds the wavelet tree for document IDs. Figure 4 illustrates an example of the wavelet tree. After sorting the concatenated text, i.e., "to be or not to be to live or to die" in Figure 3, the wavelet tree for the list of document IDs is constructed. Specifically, it is the full binary tree that classifies each document ID into either of the child nodes (0 or 1) until every child node contains a sole document ID, and keeps the order of the document IDs within a node. This enables us to find the corresponding position in a child node for a given position by using *rank* and *select* operations in $O(1)$ time [15]. Finally, it counts the document frequencies of the N-gram and the conjunction of words using the wavelet tree for every valid N-gram. Since all documents containing an N-gram is represented by a region in the list, the document frequency is counted by traversing the wavelet tree and counting the number of leaf nodes. Figure 5 shows how to list all documents containing multiple words using the wavelet tree that is built in Figure 4. Given multiple words "to" and "be" as a conjunctive query, it traverses the wavelet tree and finds leaf nodes "a" and "b" that can be reached from all the words in the query.

We coded a program that can process a set of documents to extract all valid N-grams and their N-gram IDF weight using esaxx and wat-array. Using the program, we processed English Wikipedia dump data as of Oct. 1, 2013. When

[3] https://code.google.com/p/esaxx/

[4] https://code.google.com/p/wat-array/

[5] Practically a control character is inserted between documents to identify the beginning and ending of each document.

processing it, we omitted the capitalization information. To further reduce the processing time, we introduced some techniques: $N \leq 10$ limitation, reuse of the results of overlapping N-grams and dynamic threshold of the document frequency, i.e., $\frac{1}{2000}$ of every constituent word. It still took 12 days using two high-memory (more than 60GB) machines to process 11GB of the Wikipedia corpus. Our program as well as the processed data, its demonstration page and datasets for key term extraction that are used in Section 6.1 can be found on the Web page[6].

# 6. EXPERIMENTS

We evaluated the robustness of N-gram IDF on two applications. To sum up, N-gram IDF achieved promising results for key term extraction and query segmentation. In fact, we also tried to use N-gram IDF as the bag-of-words feature in information retrieval, document clustering and document classification, though N-grams for $N > 1$ did not contribute to the performance regardless of the weighting scheme.

## 6.1 Key Term Extraction

We conducted experiments on key term extraction using Wikipedia datasets. In Wikipedia, key terms are highlighted as anchor texts that are linked to their Wikipedia articles. We hence used anchor texts as well as emphasized texts (bold texts) as correct key terms. Given a Wikipedia article, each method extracted terms from the text, calculated the weight for the terms and created a ranked list of key terms based on the weight. We then measured R-Prec, which is the precision of top $R$ terms for $R$ correct key terms. To reduce the effect of local term weighting schemes, we built a dataset using first paragraphs of randomly chosen articles. In short texts, $TF = 1$ challenge [45] exists, i.e., local term weighting scheme TF outputs 1 in most cases. We can therefore look into the performance of global term weighting schemes using short text datasets. The created dataset specifically contains 1,678 short texts in which there are 60.2 words (max: 291, min: 8) and 6.7 key terms[7] (max: 30, min: 3) on average.

Our N-gram IDF scheme can extract candidates of the key term and rank them based only on the weight. That is, dominant N-grams in a text are the candidates of the key term. The combination of TF and N-gram IDF, named N-gram TF-IDF, was employed in this experiment. As a comparative method, we employed POS tagging-based method using TF-IDF [19], which has proven to be robust. It first enumerates all noun phrases (NPs) by extracting consecutive words of NN* or JJ tags, and next measures the weight of NPs by summing the TF-IDF weight of each word. We named it TF-IDF sum with NPs. We also compared some versions of this: TF-IDF avg with NPs (using average IDF of all words) and TF-IDF with NPs (using IDF of the phrase). As the baseline, we used TF-IDF uni-gram, which only focuses on single words. All (single) stop words including numbers were removed from the candidates of the key term. Note that all global term weighting schemes except N-gram IDF are incapable of extracting key phrases by themselves. Even more, N-gram IDF is capable of extracting key terms other than simple noun phrases such as movie titles and television programs. We also evaluated POS tagging-based methods on decapitalized texts (No-Cap) to see the influence of the formality of the text.

**Table 2: Performance of key term extraction on Wikipedia first paragraph dataset.**

| Method | R-Prec |
|---|---|
| N-gram TF-IDF | 0.377 |
| TF-IDF sum with NPs | 0.386 |
| TF-IDF sum with NPs, No-Cap | 0.369 |
| TF-IDF avg with NPs | 0.367 |
| TF-IDF avg with NPs, No-Cap | 0.352 |
| TF-IDF with NPs | 0.369 |
| TF-IDF with NPs, No-Cap | 0.355 |
| TF-IDF uni-gram | 0.229 |

**Table 3: Performance of key term extraction on Wikipedia full text dataset.**

| Method | R-Prec | Prec@10 |
|---|---|---|
| N-gram TF-IDF | 0.300 | 0.358 |
| TF-IDF sum with NPs | 0.317 | 0.427 |
| TF-IDF sum with NPs, No-Cap | 0.301 | 0.409 |
| TF-IDF avg with NPs | 0.283 | 0.359 |
| TF-IDF avg with NPs, No-Cap | 0.269 | 0.333 |
| TF-IDF with NPs | 0.282 | 0.355 |
| TF-IDF with NPs, No-Cap | 0.270 | 0.341 |
| TF-IDF uni-gram | 0.154 | 0.200 |

Table 2 shows the performance results of key term extraction. TF-IDF sum with NPs achieved the best precision of 0.386, followed by N-gram TF-IDF achieving 0.377. Among POS tagging-based methods, summing the TF-IDF weight was better than averaging the weight or directly using the weight of N-grams. TF-IDF uni-gram was apparently worse than the others because it did not extract any phrases, also resulting in extracting incorrect words (uni-grams) that should compose phrases. It was surprising that N-gram TF-IDF was very competitive with the robust POS tagging-based method [19] by just using the term weight. When the capitalization information was missing, N-gram TF-IDF slightly outperformed TF-IDF sum with NPs. Taking it into consideration that the focus of text analysis has been shifting toward short and informal texts such as social media, N-gram IDF has advantages in real situations.

We also evaluated the methods on Wikipedia full text dataset containing 1,747 texts of whole articles. It contains 805.5 words (max: 16904, min: 102) and 36.8 key terms[8] (max: 999, min: 10) on average. Table 3 shows the results. Because there were many key terms in a single article, we also measured Prec@10, the precision of top 10 terms, along with R-Prec. On the Wikipedia full text dataset, the performance of N-gram TF-IDF was inferior to TF-IDF sum with NPs especially when focusing on the top 10 of ranked lists. This was mainly due to the local term weighting scheme. Whereas noun phrases are more likely to be key terms than words, words tend to occur more frequently in a long text than noun phrases. TF-IDF sum with NPs preferentially extracted key (noun) phrases because it prioritizes longer N-grams. Contrary to this, N-gram TF-IDF tried to acquire

[6] http://iwnsew.com

[7] We omitted short texts having less than three key terms.

[8] We omitted short texts having less than ten key terms.

Table 4: Performance of query segmentation on Roy et al. [40] dataset.

| Method | nDCG@5 | nDCG@10 | MAP@5 | MAP@10 | MRR@5 | MRR@10 |
|---|---|---|---|---|---|---|
| N-gram IDF | 0.730 | 0.742 | 0.900 | 0.893 | 0.582 | 0.593 |
| Mishra et al. | 0.706 | 0.737 | 0.895 | 0.892 | 0.529 | 0.542 |
| Mishra et al. with Wikipedia | 0.725 | 0.750 | 0.907 | 0.902 | 0.561 | 0.571 |
| PMI-Q | 0.716 | 0.736 | 0.898 | 0.892 | 0.567 | 0.577 |
| PMI-W | 0.670 | 0.707 | 0.860 | 0.863 | 0.493 | 0.506 |
| Unsegmented | 0.655 | 0.689 | 0.852 | 0.854 | 0.465 | 0.481 |
| Human A | 0.728 | 0.746 | 0.904 | 0.899 | 0.575 | 0.585 |
| Human B | 0.727 | 0.747 | 0.903 | 0.898 | 0.567 | 0.577 |
| Human C | 0.717 | 0.744 | 0.899 | 0.896 | 0.543 | 0.555 |
| BQV | 0.765 | 0.768 | 0.927 | 0.914 | 0.673 | 0.680 |

more key words since it fairly gives the weight to N-grams of any length, sacrificing the precision. In fact, the number of key words extracted with N-gram TF-IDF was 7,701, exceeding 7,030 obtained with TF-IDF sum with NPs. To apply N-gram IDF to long texts where not a few terms occur more than once, the local term weighting scheme should be well designed.

## 6.2 Query Segmentation

Query segmentation is another application of N-gram IDF. In this evaluation, we used an IR-based Web search query segmentation dataset [40]. The dataset contains 13,959 Web documents, 500 test queries and their qrels (query-relevance sets). The qrels were created by three human experts grading the query-document relevance score by 2 (highly relevant), 1 (relevant) or 0 (irrelevant). The average relevance score of the three experts was used as the gold standard in our evaluation. Given a query (e.g., "larry the lawnmower tv show"), each method segmented it into some words and phrases (e.g. "larry the lawnmower" and "tv show"). The words and phrases were then used to force the search system to match them exactly in documents. This is a Boolean query using double quotes in general Web search engines. After collecting documents that satisfy the Boolean query, we simply calculated the conventional TF-IDF weight for each document. Here, IDF was computed using the dataset documents. Because selecting which words or phrases should be quoted is a difficult problem, Roy et al. [40] measured evaluation scores for all possible quoted queries given a segmentation result obtained with each method. We followed their manner in our evaluation.

The dataset includes some results obtained with comparative methods: Mishra et al. [28], Mishra et al. with Wikipedia titles, PMI using query logs (PMI-Q) and PMI using Web documents (PMI-W). The thresholds of PMI-Q and PMI-W were adjusted by Roy et al. using their development set. It also provides three segmentation results by humans. Note that they were different from those who rated the qrels. We measured evaluation scores for them as well as unsegmented queries and (probably) the best quoted version of the queries (BQV)[9].

N-gram IDF was computed using the dataset documents because many queries and documents were informal and N-gram IDF computed in Wikipedia did not cover some N-grams in the dataset. Processing 283MB of the dataset documents took 40 minutes. We employed the same evaluation metrics as Roy et al.: normalized Discounted Cumulative Gain (nDCG), Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) for top 5 and 10 search results. Because computing MAP and MRR requires binary values of the qrel score, we regarded 2 and 1 as relevant when computing MAP, and only 2 as relevant for MRR.

Table 4 shows the performance results[10]. N-gram IDF was competitive with well-adjusted methods such as Mishra et al. with Wikipedia and PMI-Q, or human segmentation results, in all evaluation metrics. Let us recollect that N-gram IDF just has the weight of each N-gram like conventional IDF. Mishra et al. with Wikipedia leverages the human knowledge of Wikipedia and PMI-Q requires query logs and the threshold adjustment. Against these methods, N-gram IDF was able to achieve competitive performance by simply selecting dominant N-grams based on the weight. From these results, the simplicity and robustness of N-gram IDF were demonstrated.

# 7. CONCLUSIONS

This paper first time ever revealed the relationship between IDF and information distance. Specifically, the IDF of a term is equal to the distance between the term and the empty string in the space of information distance where the Kolmogorov complexity is approximated using Web documents and the Shannon-Fano coding. Our findings are helpful when designing a global term weighting scheme because the information distance can be regarded as the term weight. Based on our findings, we also proposed a global term weighting scheme, N-gram IDF, by incorporating IDF and MED, a universal information distance-based metric for measuring the word compositionality. N-gram IDF is able to handle N-grams of any $N$ in a sole theoretical scheme. It enables us to compare the weight of words and phrases, and thus, select dominant N-grams among overlapping ones. We demonstrated the simplicity and robustness of N-gram IDF on key term extraction and Web search query segmentation tasks. N-gram IDF was able to achieve competitive perfor-

[9] BQV in the literature [40] may not be the best because it does not consider partly overlapping phrases such as "free solitaire" and "solitaire card games" for query "play free solitaire card games."

[10] Evaluation scores of the comparative methods were different from those reported by Roy et al. [40] because we used the TF-IDF weight that was computed using the dataset documents. We observed the same tendencies that did not contradict their results.

mance with state-of-the-art methods designed for each task using extra resources and efforts.

Our future work includes the development of approximation methods to shorten the processing time for handling large document sets. As the corpus size (the number of documents $|D|$) grows, the number of valid N-grams also increases linearly. The processing time for computing the N-gram IDF weight for all valid N-grams therefore becomes roughly $O(|D|^2)$. Fortunately, Bu et al. [7] reported that MED of a term was rather well measured by focusing on documents whose topics were related to the term. In order to manage the big text data and upgrade the quality of the weight, we plan to use small and biased document sets for measuring MED. Another future work is to handle texts written in languages without spaces between words such as Japanese and Chinese. Possible approaches to solve this problem are two: designing character-level N-gram IDF or introducing unsupervised word segmentation techniques. We will explore the possibility of both approaches.

## 8. ACKNOWLEDGMENTS

This work was supported in part by CPS-IIP Project (Integrated Platforms for Cyber-Physical Systems to Accelerate Implementation of Efficient Social Systems) in the research promotion program for national level challenges “research and development for the realization of next-generation IT platforms” and by the Grant-in-Aid for Scientific Research (A)(26240013) by MEXT, Japan.

## 9. REFERENCES

[1] M. I. Abouelhoda, S. Kurtz, and E. Ohlebusch. Replacing Suffix Trees with Enhanced Suffix Arrays. *Journal of Discrete Algorithms*, 2(1):53–86, Mar. 2004.

[2] A. Aizawa. An Information-Theoretic Perspective of TF-IDF Measures. *Information Processing and Management*, 39(1):45–65, Jan. 2003.

[3] C. H. Bennett, P. Gács, M. Li, P. M. Vitányi, and W. H. Zurek. Information Distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423, July 1998.

[4] A. Blumer, J. Blumer, D. Haussler, R. M. McConnell, and A. Ehrenfeucht. Complete Inverted Files for Efficient Text Retrieval and Analysis. *Journal of the ACM*, 34(3):578–595, July 1987.

[5] A. Bookstein and D. R. Swanson. Probabilistic Models for Automatic Indexing. *Journal of the American Society for Information Science*, 25(5):312–318, Sept./Oct. 1974.

[6] G. Bouma. Normalized (Pointwise) Mutual Information in Collocation Extraction. In *Proceedings of International Conference of the German Society for Computational Linguistics and Language Technology (GSCL)*, pages 31–40, Sep./Oct. 2009.

[7] F. Bu, X. Zhu, and M. Li. Measuring the Non-compositionality of Multiword Expressions. In *Proceedings of International Conference on Computational Linguistics (COLING)*, pages 116–124, Aug. 2010.

[8] K. W. Church and W. A. Gale. Poisson Mixtures. *Natural Language Engineering*, 1(2):163–190, 1995.

[9] K. W. Church and P. Hanks. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, 16(1):22–29, Mar. 1990.

[10] R. L. Cilibrasi and P. M. Vitányi. Clustering by Compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, Apr. 2005.

[11] R. L. Cilibrasi and P. M. Vitányi. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383, Mar. 2007.

[12] J. F. da Silva and J. G. P. Lopes. A Local Maxima Method and a Fair Dispersion Normalization for Extracting Multi-word Units from Corpora. In *Proceedings of Meeting on Mathematics of Language (MOL)*, pages 369–381, July 1999.

[13] G. Dias. Mining Textual Associations in Text Corpora. In *Proceedings of ACM SIGKDD Workshop on Text Mining*, pages 20–23, Aug. 2000.

[14] B. C. Fung, K. Wang, and M. Ester. Hierarchical Document Clustering Using Frequent Itemsets. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, pages 59–70, May 2003.

[15] T. Gagie, G. Navarro, and S. J. Puglisi. New Algorithms on Wavelet Trees and Applications to Information Retrieval. *Theoretical Computer Science*, 426–427:25–41, Apr. 2012.

[16] W. R. Greiff. A Theory of Term Weighting Based on Exploratory Data Analysis. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 11–19, Aug. 1998.

[17] R. Grossi, A. Gupta, and J. S. Vitter. High-Order Entropy-Compressed Text Indexes. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 841–850, 2003.

[18] S. P. Harter. A Probabilistic Approach to Automatic Keyword Indexing. Part I. On the Distribution of Specialty Words in a Technical Literature. *Journal of the American Society for Information Science*, 26(4):197–206, July/Aug. 1975.

[19] K. S. Hasan and V. Ng. Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art. In *Proceedings of International Conference on Computational Linguistics (COLING)*, pages 365–373, Aug. 2010.

[20] D. Hiemstra. A Probabilistic Justification for Using TF×IDF Term Weighting in Information Retrieval. *International Journal on Digital Libraries*, 3(2):131–139, Aug. 2000.

[21] K. S. Jones. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28:11–21, 1972.

[22] A. Kolmogorov. On Tables of Random Numbers. *Sankhyā Ser. A*, 25:369–376, 1963.

[23] M. Lan, C. L. Tan, J. Su, and Y. Lu. Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):721–735, Apr. 2009.

[24] R. Landauer. Irreversibility and Heat Generation in the Computing Process. *IBM Journal of Research and Development*, 5(3):183–191, July 1961.

[25] M. Li and P. M. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, Berlin, 3rd edition, 2009.

[26] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, 2008.

[27] D. Metzler. Generalized Inverse Document Frequency. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 399–408, Oct. 2008.

[28] N. Mishra, R. S. Roy, N. Ganguly, S. Laxman, and M. Choudhury. Unsupervised Query Segmentation Using only Query Logs. In *Proceedings of International World Wide Web Conference (WWW)*, pages 91–92, Mar./Apr. 2011.

[29] K. Narisawa, S. Inenaga, H. Bannai, and M. Takeda. Efficient Computation of Substring Equivalence Classes with Suffix Arrays. In *Proceedings of Symposium on Combinatorial Pattern Matching (CPM)*, pages 340–351, July 2007.

[30] D. Okanohara and J. Tsujii. Text Categorization with All Substring Features. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, pages 838–846, Apr./May 2009.

[31] C. Orăsan, V. Pekar, and L. Hasler. A Comparison of Summarisation Methods Based on Term Specificity Estimation. In *Proceedings of International Conference on Language Resources and Evaluation (LREC)*, pages 1037–1041, May 2004.

[32] K. Papineni. Why Inverse Document Frequency? In *Proceedings of Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1–8, June 2001.

[33] P. Pecina. An Extensive Empirical Study of Collocation Extraction Methods. In *Proceedings of ACL Student Research Workshop*, pages 13–18, June 2005.

[34] J. W. Reed, Y. Jiao, T. E. Potok, B. A. Klump, M. T. Elmore, and A. R. Hurson. TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams. In *Proceedings of International Conference on Machine Learning and Applications (ICMLA)*, pages 258–263, Dec. 2006.

[35] J. D. M. Rennie and T. Jaakkola. Using Term Informativeness for Named Entity Detection. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 353–360, Aug. 2005.

[36] S. Robertson. Understanding Inverse Document Frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520, 2004.

[37] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of Text Retrieval Conference (TREC)*, pages 109–126, 1994.

[38] T. Roelleke and J. Wang. TF-IDF Uncovered: A Study of Theories and Probabilities. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 435–442, July 2008.

[39] F. Rousseau and M. Vazirgiannis. Graph-of-word and TW-IDF: New Approach to Ad Hoc IR. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 59–68, Oct./Nov. 2013.

[40] R. S. Roy, N. Ganguly, M. Choudhury, and S. Laxman. An IR-based Evaluation Framework for Web Search Query Segmentation. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 881–890, Aug. 2012.

[41] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[42] G. Salton, A. Wong, and C.-S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[43] P. Schone and D. Jurafsky. Is Knowledge-Free Induction of Multiword Unit Dictionary Headwords a Solved Problem? In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 100–108, June 2001.

[44] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1470–1477, Oct. 2003.

[45] M. Timonen. *Term Weighting in Short Documents for Document Categorization, Keyword Extraction and Query Expansion*. PhD thesis, University of Helsinki, 2013.

[46] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok. Interpreting TF-IDF Term Weights as Making Relevance Decisions. *ACM Transactions on Information Systems*, 26(3):13:1–13:37, June 2008.

[47] M. Yamamoto and K. W. Church. Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus. *Computational Linguistics*, 27(1):1–30, Mar. 2001.

[48] W. Zhang, T. Yoshida, X. Tang, and T.-B. Ho. Improving Effectiveness of Mutual Information for Substantival Multiword Expression Extraction. *Expert Systems with Applications*, 36(8):10919–10930, Oct. 2009.